EVALUATION OF A HIGHLY SKEWED PROPELLER FOR A NAVAL AUXILIARY (AO-177)

by

RICHARD L. JAMISON
LIEUTENANT, U.S. NAVY
S.B., Massachusetts Institute of Technology
(1976)

SUBMITTED TO THE DEPARTMENT OF OCEAN
ENGINEERING IN PARTIAL FULFILLMENT
OF THE DEGREE OF OCEAN ENGINEER
AND THE THE DEGREE OF MASTER OF
SCIENCE IN NAVAL ARCHITECTURE
AND MARINE ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June, 1982

EVALUATION OF A HIGHLY SKEWED PROPELLER FOR A NAVAL AUXILIARY (AO-177)

by

Richard L. Jamison

Submitted to the Department of Ocean Engineering on May 7, 1982, in partial fulfillment of the requirements for the degree of Ocean Engineer and the degree of Master of Science in Naval Architecture and Marine Engineering.

ABSTRACT

The cavitation performance of a seven-bladed, highly skewed propeller for a Naval auxiliary (AO-177) is evaluated using lifting surface numerical hydrodynamic methods. An important contribution to accurate predition of cavitation performance is shown to be an accurate model of the effective wake.

A new model of the effective wake is described. A program to calculate unsteady time-averaged, but not circumferentially averaged, field point velocities is presented. The propeller-induced velocities, along with the original nominal wake, are combined with Huang's axisymmetric effective wake scheme in pie-shaped wake segments to determine the effective wake.

The method is used to predict severe cavitation extent. This prediction is confirmed by SSPA experiments.

Thesis Supervisor: David V. Burke

Title: Professor of Naval Architecture

ACKNOWLDGEMENT

The author wishes to express his gratitude to the Navy for supporting this thesis. In particular, M. Wilson and R. Boswell of DTNSRDC and A. Chase of NAVSEA provided the needed raw data and background information in a timely manner and expressed strong support by way of their interest.

Thanks are extended to Professor D. V. Burke, who served as thesis advisor and who also provided the original motivation for the project, and to Professors J. E. Kerwin and R. Van Houten, whose experience and physical insight invariably got the project moving again each time it bogged down.

I would also like to express my appreciation to my wife, Beverly, not only for her support of our home, but also for her technical assistance in teaching me programming techniques and debugging tricks.

Finally, I wish to thank my daughter, Ruth, just for being Ruth.

TABLE OF CONTENTS

I. INTRODUCTION

The David W. Taylor Naval Ship Research and Development Center has recently designed a seven-bladed, highly skewed propeller for a Naval Auxiliary (AO-177). This design is instructive concerning the state of the art in computer-aided propeller design.

The history of the propeller design is chronicled in reports by Boswell (1), Valentine and Chase (2), and Hendrican and Remmers (3). Computer-aided propeller applications included lifting line theory, used to determine the radial load distribution and the radial hydrodynamic pitch angle, and lifting surface theory, used to determine the final geometric pitch distribution, camber distribution, and final propeller offsets, including fillets, trailing and leading edge details, additional thickness added to the trailing edge, tip geometry, rake, and hub details. Cavitation performance was predicted by the method of Burrill and Emerson (4) and was also evaluated by experiment.

However, sea trials of the AO-177 indicated a severe unsteady propeller force problem. Inspection of the propeller indicated that it was cavitating significantly worse than had been predicted. The Burrill and Emerson method, originally designed for

four-to-six-bladed propellers with an expanded area ratio of 0.60, did not extend well to the seven-bladed AO-177 propeller with an expanded area ratio of 0.77, even though both were for merchant-type hulls. Further, scale effects could have thrown off the model tests.

II. IMPACT OF EFFECTIVE WAKE

The AO-177 propeller performance was evaluated at MIT using some numerical propeller analysis computer programs developed at MIT. Kerwin and Lee (5) describe MIT-PUF-2, a computer program to calculate forces, steady and unsteady, generated by a propeller. MIT-FPV, a steady field point velocity program, is described in a report by Min (6). Lee (7) reports on MIT-PUF-3, a computer program to predict steady and unsteady propeller cavitation extent and volume. All three programs model the propeller as a grid of discrete vortex segments. The wake is also modelled as a vortex grid in each program.

Cavitation performance calculations, more than propeller force and induced velocity calculations, are highly sensitive to the wake data used as input. Experience at MIT and elsewhere has shown that using the nominal wake data to approximate inflow conditions, while acceptable for propeller force and induced velocity calculations, is not acceptable for cavitation performance calculations. The wake velocities measured behind a model in the absence of any propeller (the nominal wake) must be modified to account for change in boundary conditions imposed by the presence of a propeller. This new, modified wake is generally called the "effective wake." There is, unfortunately, no

general method for anticipating the effective wake generated by a particular propeller in a general nominal wake.

Huang and others (8,9) have developed both theory and numerical schemes for calculating the effective wake of a propeller in an axisymmetric nominal wake. Det Norske Veritas has attempted to adapt Huang's method to non-axisymmetric wakes, such as surface ship wakes, by dividing the nominal wake into pie-shaped segments and then applying the axisymmetric wake calculation within each pie segment, using the nominal wake and the steady propeller induced velocities. This approach is an improvement over blindly applying the axisymmetric effective wake calculation to a surface ship's wake, but, owing to the very sharp AO-177 wake (even by surface ship standards), it was judged that some further refinement of Huang's method was required.

Huang's method assumed an axisymmetric nominal wake and a steady propeller-induced velocity field. The Det Norske Veritas method assumed a general nominal wake and a steady propeller-induced velocity field. Short of a completely general effective wake calculation, the most obvious refinement was to assume a general nominal wake, but allow an unsteady propeller-induced velocity field. This was the method

used for this paper.

The current method of effective wake calculations can be summarized in the following steps:

1) Use the nominal wake and propeller data to calculate the unsteady propeller-induced velocity field at representative points both inside and outside the propeller disk.

2) Divide the nominal wake into small pie-shaped segments as in the Det Norske Veritas method.

3) Calculate the effective wake velocities within each pie segment using the nominal wake velocities and the unsteady propeller-induced velocities, assuming that each pie segment acts as though it were part of an axisymmetric wake with the same radial velocity distribution.

III. METHOD OF EFFECTIVE WAKE CALCULATION

The two new aspects of the current effective wake calculations are the propeller-induced unsteady field point velocity (UFPV) calculation and the use of these time-averaged but not circumferentially-averaged velocities in a pie segment nominal wake modification calculation (PIEWAKE). Each of these two new methods will be illustrated by a typical calculation.

The unsteady field point velocity (UFPV) program is basically a generalization of the steady field point velocity program (MIT-FPV) by Min. (6). The unsteady version uses the full wake lattice arrangement also employed in MIT-PUF-2. Program inputs are the FILE14 DATA file output from MIT-PUF-2 and interactive directions. Program outputs are either single point calculations directed to the user's terminal or a data file to be used as input into PIEWAKE, for effective wake calculations.

UFPV first reads the FILE14 DATA input, containing mostly propeller geometry, singularity strengths, and singularity geometry. It should be noted that the number of propeller revolution time steps specified in MIT-PUF-2 must be divisible by the number of propeller blades . For effective wake calculations, the wake input to MIT-PUF-2 ought strictly to be the effective wake; however, using the nominal wake to calculate the propeller-induced velocities

is a good approximation.

The user next inputs field point coordinates. The field point position may be entered in cylindrical coordinates or in terms of the rotating propeller and wake grids.

The velocity induced by one blade at each time step is then calculated. At each time step, the vortex lattice is assigned the strengths corresponding to the time step of the propeller rotation. The velocity induced by each element of the vortex grid, both blade and wake, is then summed to obtain the total velocity induced by one blade in that particular angular position.

The velocities induced by each blade of the propeller are then summed to produce the total propeller-induced velocity for that position of the propeller. Velocities are calculated for different propeller positions and then harmonically analyzed. The zeroeth harmonic, then, is the time-averaged propeller-induced velocity for that field point. (By averaging all points at a given radius we could obtain the circumferentially-averaged, time-averaged, propeller-induced velocity, the output of the steady MIT-FPV.)

For effective wake calculations, such velocities are computed for up to 60 points per radius for radii varying from propeller hub to 1.7 times the propeller radius.

Between the propeller hub and propeller tip, the velocities are calculated at the leading edge panel on the propeller grid.

For the AO-177 propeller analysis, 56 points per radius for eight radii were used to create a file of unsteady field point velocities.

PIEWAKE is a program to perform Huang's axisymmetric effective wake contraction calculations in each of many pie-shaped segments of the wake. It uses nominal wake data and unsteady propeller-induced velocity data from UFPV as inputs to produce an effective wake velocity data file as output.

After reading nominal wake data and the induced velocity field data, PIEWAKE extrapolates the nominal wake data to the hub. The innermost radius of the induced velocity field is taken as the hub. (UFPV had taken, as its innermost radius, the hub radius used in MIT-PUF-2 and passed to UFPV by way of FILE14 data file.)

In each pie segment, Huang's method is used to calculate an effective velocity and the effective radius corresponding to that effective velocity. In the numerical form, this method is iterative in nature.

A finite difference equation, given in reference (9), is used to calculate the effective velocity. Given the

nominal wake at radii (assumed, for the first iteration, to be the nominal radii) in the presence of a propeller, an effective velocity can be calculated if an effective velocity at the next radius away from the hub is known. The effective velocity at the outermost radius is assumed to be equal to the nominal velocity there, enabling all effective velocities to be calculated.

A second finite difference equation, also given in reference (9), is used to calculate the effective radius corresponding to the set of effective velocities just calculated. This equation uses nominal radii, nominal velocities, and apparent velocities. (An apparent velocity is the sum of the effective velocity and the propeller-induced velocity at a point.) Given one effective radius, the next radius away from the hub can be calculated. Since the innermost radius of all velocity fields is set at the hub radius, all effective radii can be determined sequentially from innermost to outermost. These radii can then be used in the finite difference equation to calculate the next iteration of effective velocities. PIEWAKE iterates five times for each pie segment.

In addition to the effective velocity wake field, PIEWAKE also computes the mean velocity, the volumetric mean velocity of the nominal wake, the volumetric mean velocity of the effective wake, and the effective blockage.

The effective velocity field then then be processed

and used by MIT-PUF-3 as any other wake field.

IV. AO-177 EFFECTIVE WAKE CALCULATIONS

Appendix I contains plots of nominal, induced, and effective velocities varying circumferentially at several specified radii. Appendix II contains the same information, except that the nominal and effective velocities are normalized on the respective volumetric mean velocity. These plots confirm that the modifications to the nominal wake outlined in Chapter III are indeed reasonable. Several salient conclusions can be drawn from these plots:

1) Effective velocity modifications of the nominal velocities are in the expected direction. Physically, the imposition of the propeller on the nominal wake draws more water into the slipstream, thus increasing the velocities. This "faster water" will come from radii greater than the field point being sampled; in other words, the wake will seem to "contract."

2) Effective velocity modifications of the nominal wake velocities are greatest at lower radii. The wake "contraction" will seem to concentrate here.

3) In general, the wake peak is more narrow in the effective wake than in the nominal wake. In the case of the A0-177 nominal wake, the existence in the presence of a propeller of a wake peak as sharp as that in the nominal wake is counter-intuitive. Therefore, a decrease in the wake peak is to be expected.

4) The normalized effective velocities at 0 degrees

are not appreciably different from that of the nominal wake. This is disappointing and was not expected. It may be a result of radial vorticity, ignored by the axisymmetric effective wake calculation performed in each pie segment. This assumption is least appropriate in the 0 degree region of the wake. A more general effective wake calculation scheme may be required to obtain better results in this region.

Results of MIT-PUF-3 cavitation performance of the AO-177 propeller are shown in Appendices III and IV. As an indicator of the influence of effective wake modifications to the nominal wake, MIT-PUF-3 was run twice, once with the nominal wake as input and once with the effective wake (the more accurate boundary condition) as input. Appendix III shows cavitation performance given the nominal wake as input while Appendix IV shows cavitation performance given the effective wake as input. In general, the plots show that the cavitation extent is not changed much, although cavitation performance is better in the effective wake at the lower radii where, as has been shown, the effective wake modifications are the greatest. At the outer radii, where the effective wake modification is the least significant, the cavitation performance is least changed between the nominal wake and effective wake cases.

Appendix V contains a graph comparing cavitation extent of MIT-PUF-3 output given nominal wake input against that given effective wake input. It plots the length of

cavitation on the seventh radial cavitation panel (about the .8 propeller radius) non-dimensionalized on the local chord length as the blade rotates through 360 degrees. This graph indicates that the effective wake modification did not reduce cavitation extent on a blade, a result noted earlier. However, a larger contrast between the nominal wake and effective wake cases is shown in Appendix VI which graphs the cavitation volume on a blade in the two cases. The volume is non-dimensionalized on the cube of the propeller radius. This graph shows a drop of approximately 35% of peak cavitation volume. This change in cavity volume demonstrates the sensitivity of cavitation performance to small changes in the wake profile. This confirms that efforts to refine effective wake calculations were indeed justified.

The effective wake cavitation performance indicated by MIT-PUF-3 compares reasonably well with experiments performed by the SSPA propeller tunnel. These experiments measured cavitation performance of the AO-177 propeller by photographing model tests. While no cavitation volume results were obtained, cavitation extent could be compared with MIT-PUF-3 predictions. Despite reductions of the effective wake calculations compared with nominal wake calculations, MIT-PUF-3 still tended to predict greater cavitation than the experiments indicated, particularly at the middle radii. This seems to indicate that the effective wake modifications, while tending to the proper direction, were not large enough. This is consistent with

the conclusions from comparing the two wakes that a more general effective wake scheme including radial vorticity in high wake regions is required.

V. CONCLUSIONS

1) Calculation of an effective wake using Huang's axisymmetric method in pie-shaped segment with unsteady propeller-induced velocities is a reasonable approximation. It still tends to underestimate the effective wake modification to the nominal wake in the vicinity of 0 degrees. To correct this, a more general effective wake calculation scheme including radial vorticity effects may be required.

2) The current effective wake calculation scheme can be used to predict cavitation performance for highly skewed propellers with up to seven blades.

3) Cavitation performance is indeed sensitive to small changes in the wake profile. Seemingly small improvements in effective wake calculations yield significantly more accurate cavitation performance predictions.

VI. RECOMMENDATIONS

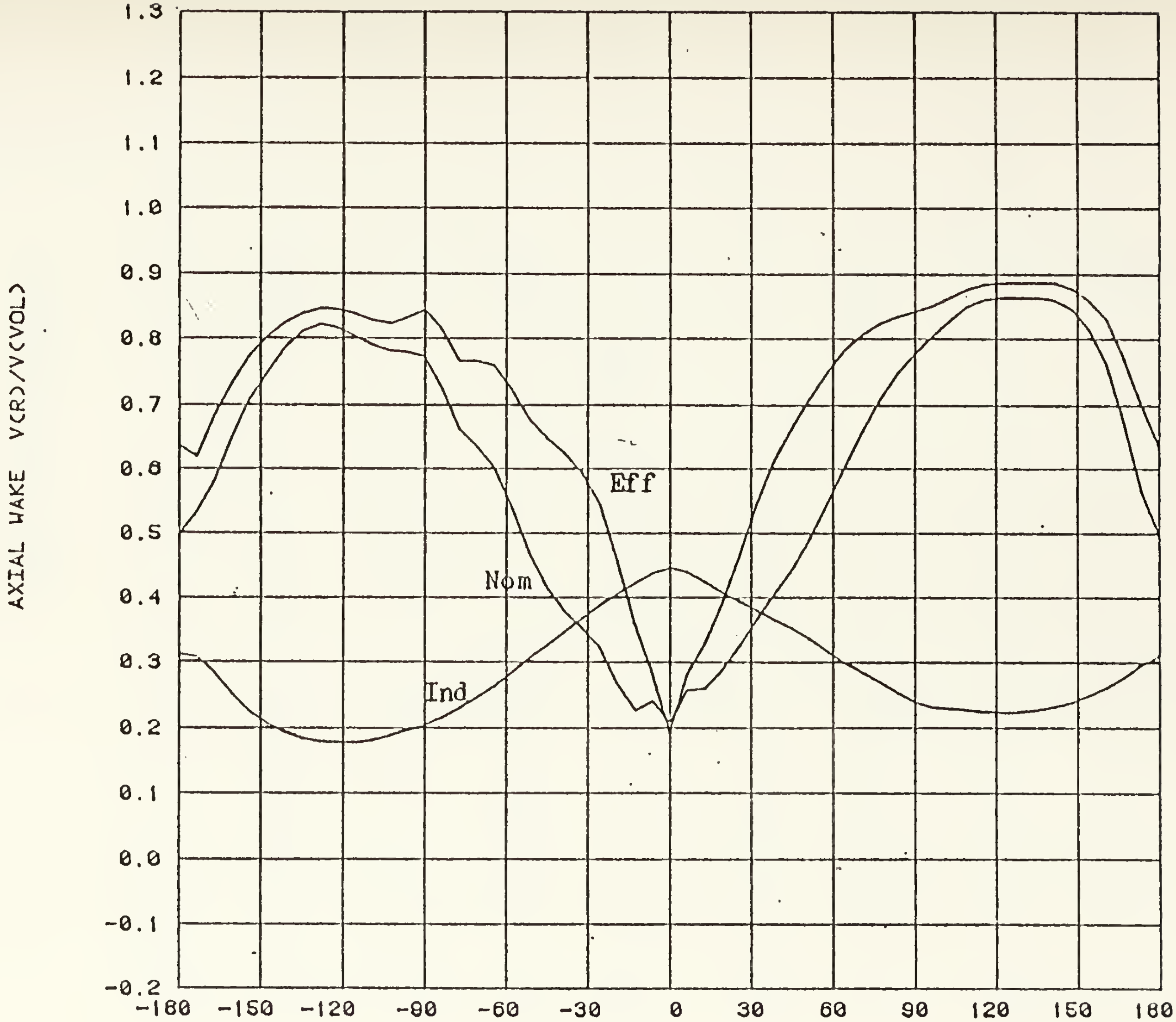
1) A more general effective wake calculation scheme, including the effects of radial vorticity, is required.

2) The computational time (and hence cost) of the unsteady field point velocity program (UFPV) can probably be cut without significant sacrifice of accuracy by decreasing the number of points calculated per radius. Appendix I shows that the variation of induced velocity with rotation is slight and smooth. Calculation of fewer points and estimation of the remaining points using harmonic analysis would probably not reduce accuracy.

** APPENDIX I **

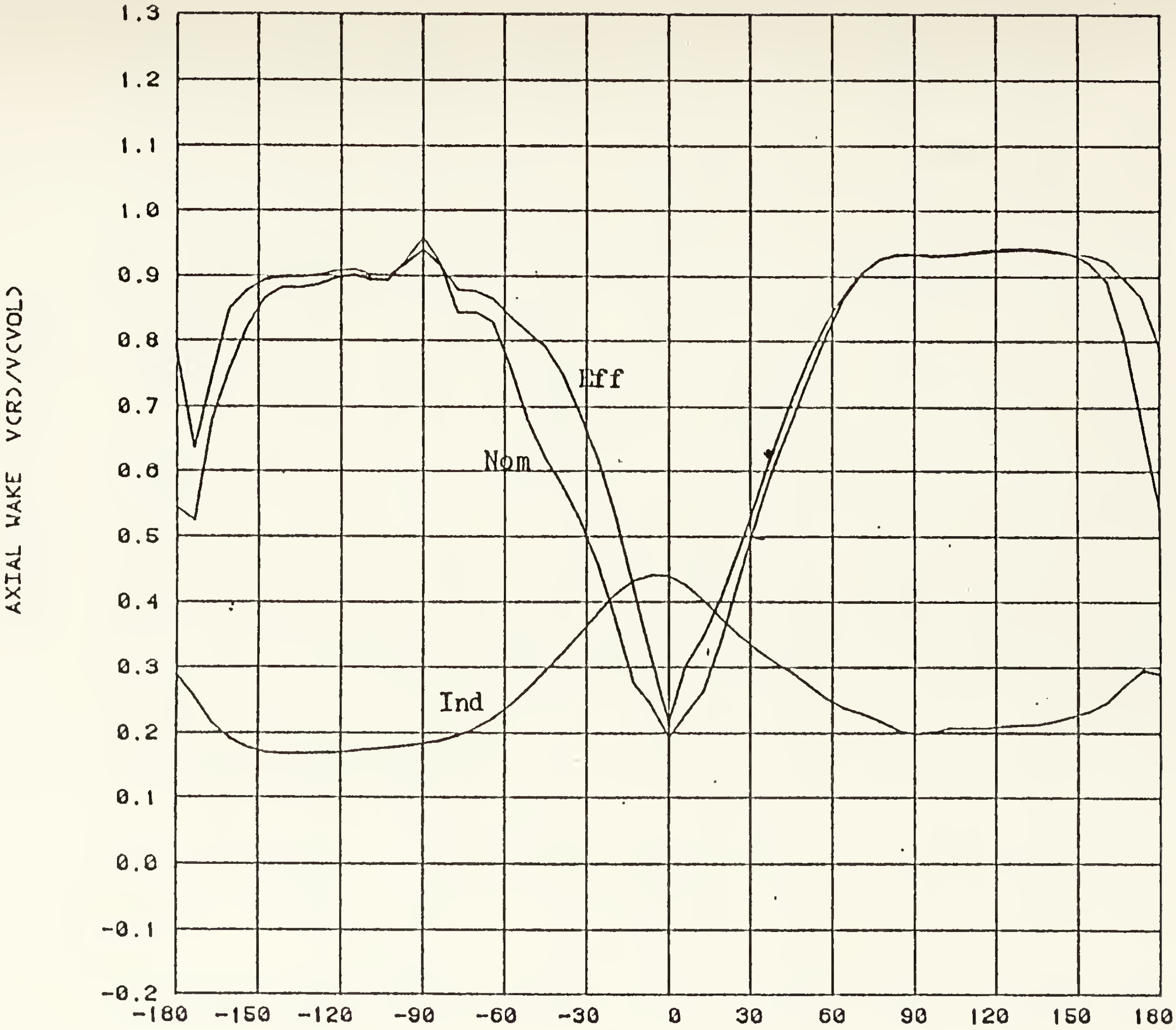
Nominal, Induced, and Effective Velocities

	V(VOL)
Nom	1.0
Ind	1.0
Eff	1.0

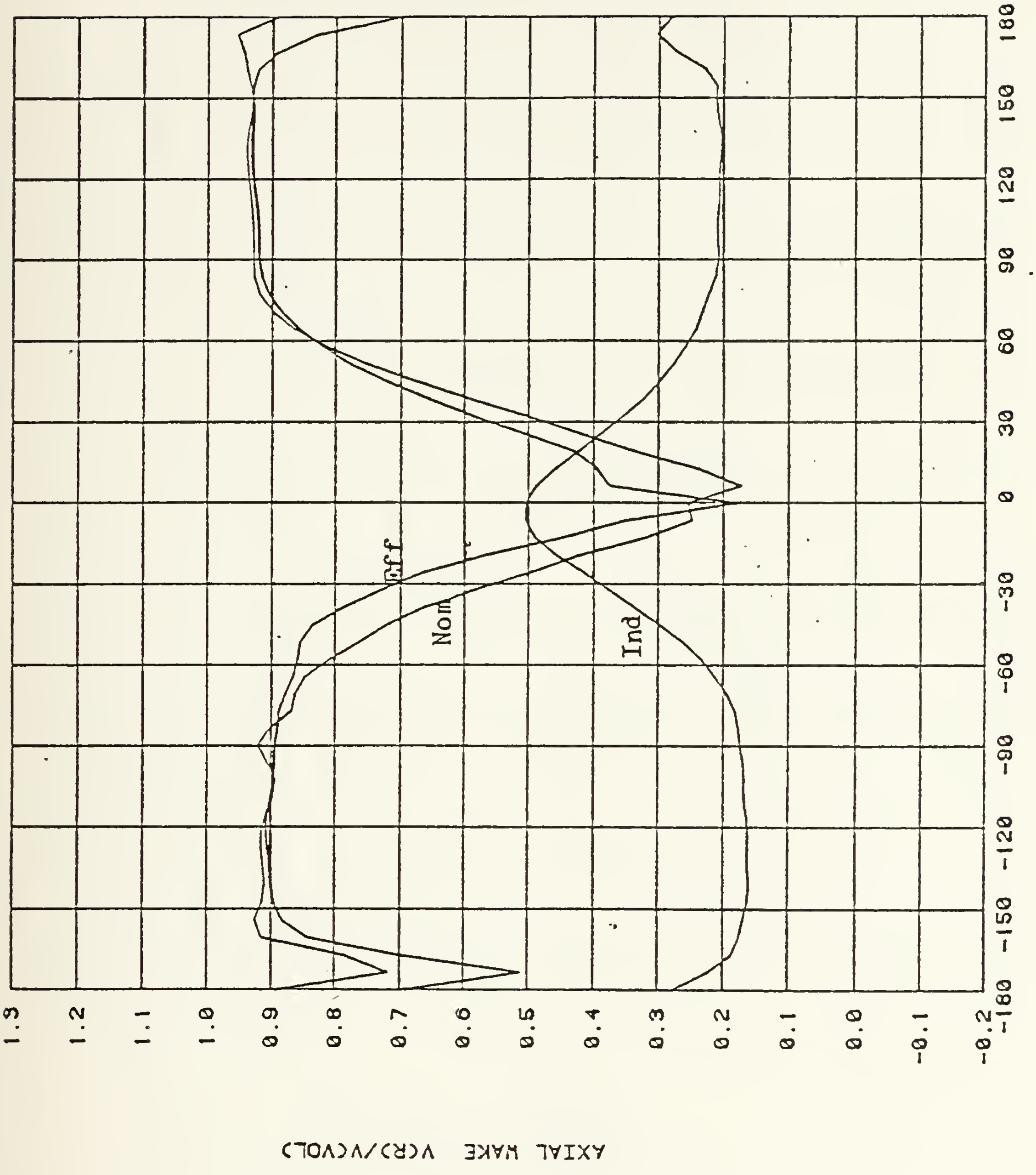


NOM, IND, EFF VELOCITIES, .40 RADIUS

	V(VOL)
Nom	1.0
Ind	1.0
Eff	1.0

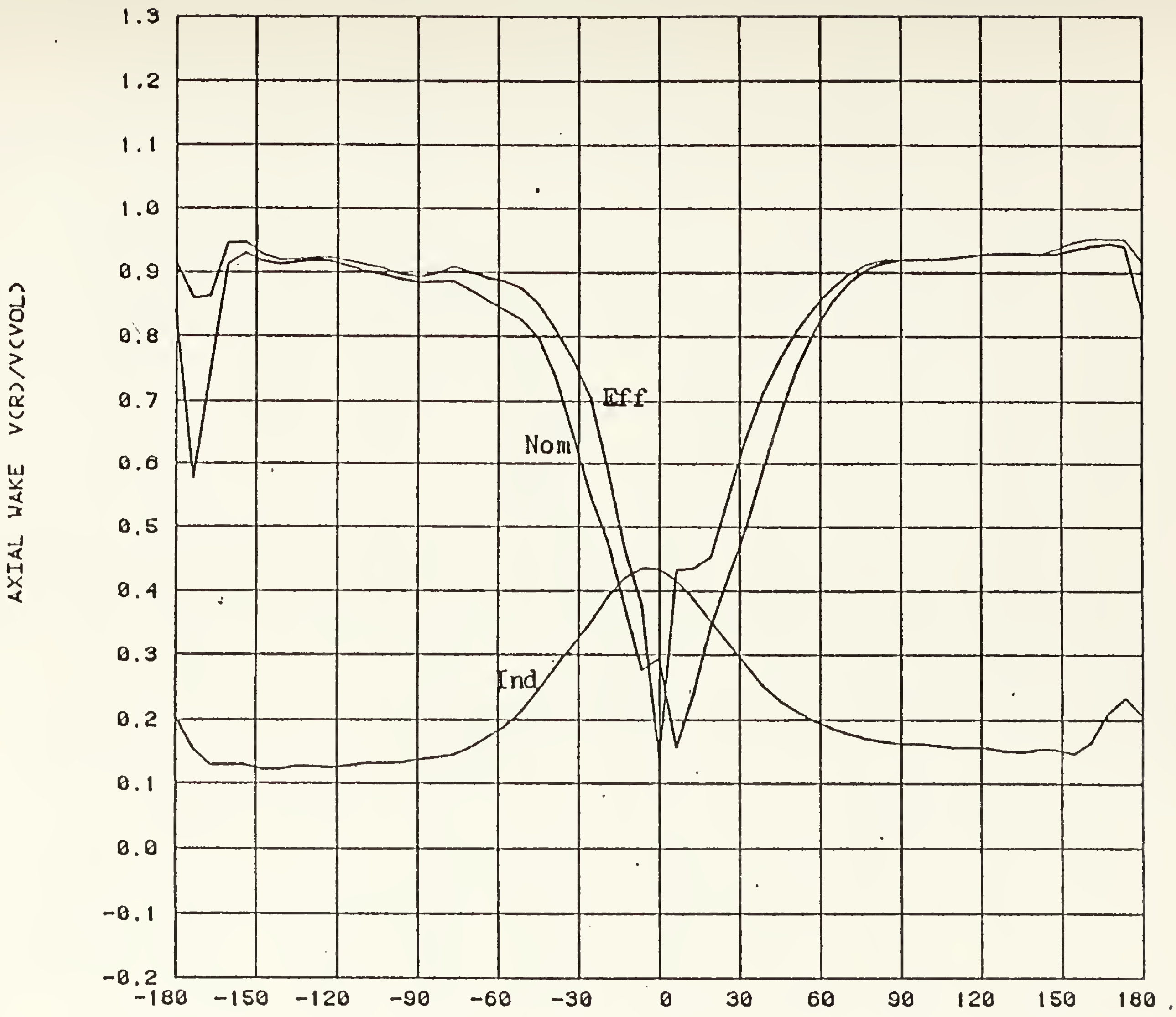


NOM, IND, EFF VELOCITIES, .60 RADIUS



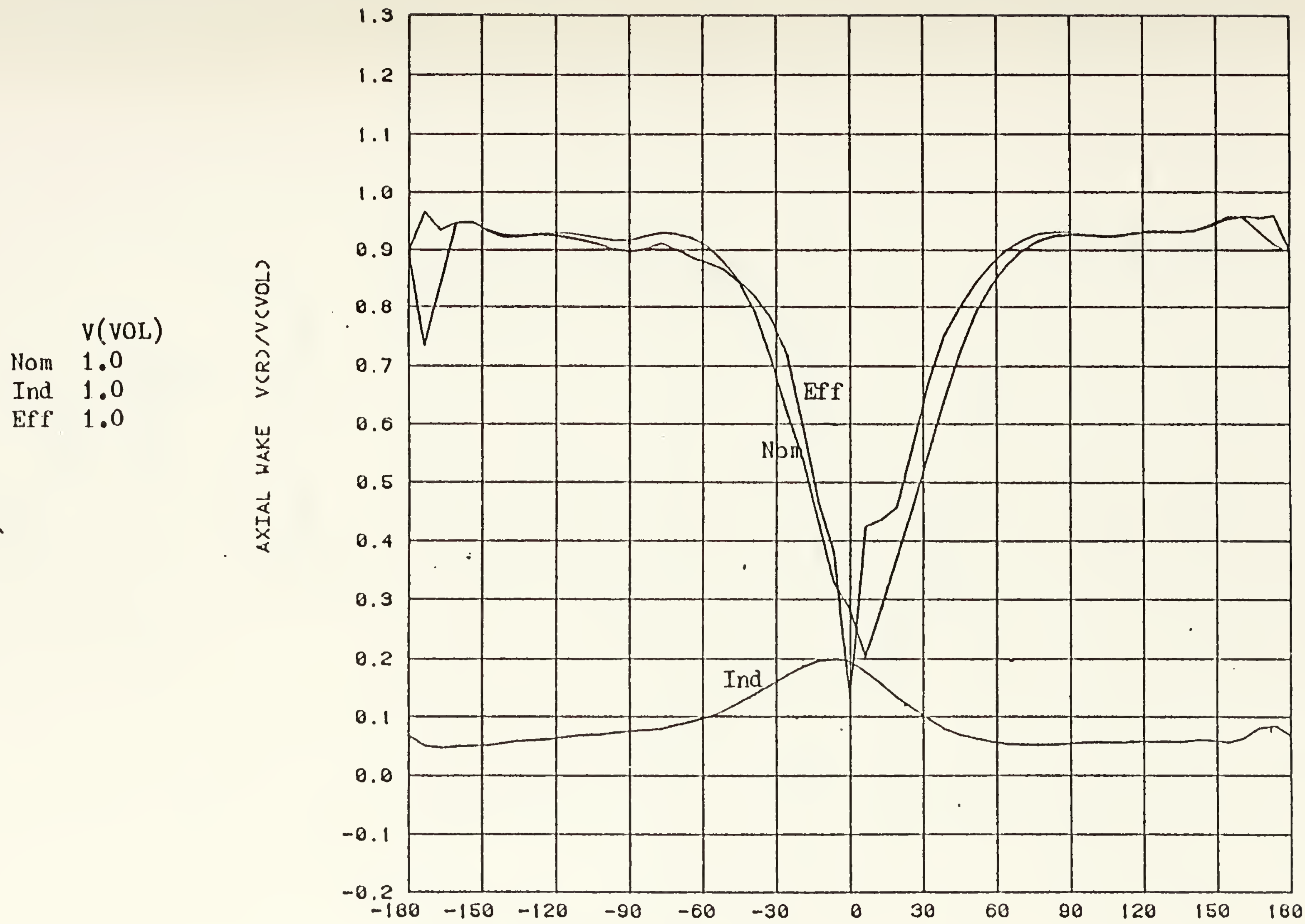
NOM, IND, EFF VELOCITIES, .70 RADIUS

	V(VOL)
Nom	1.0
Ind	1.0
Eff	1.0

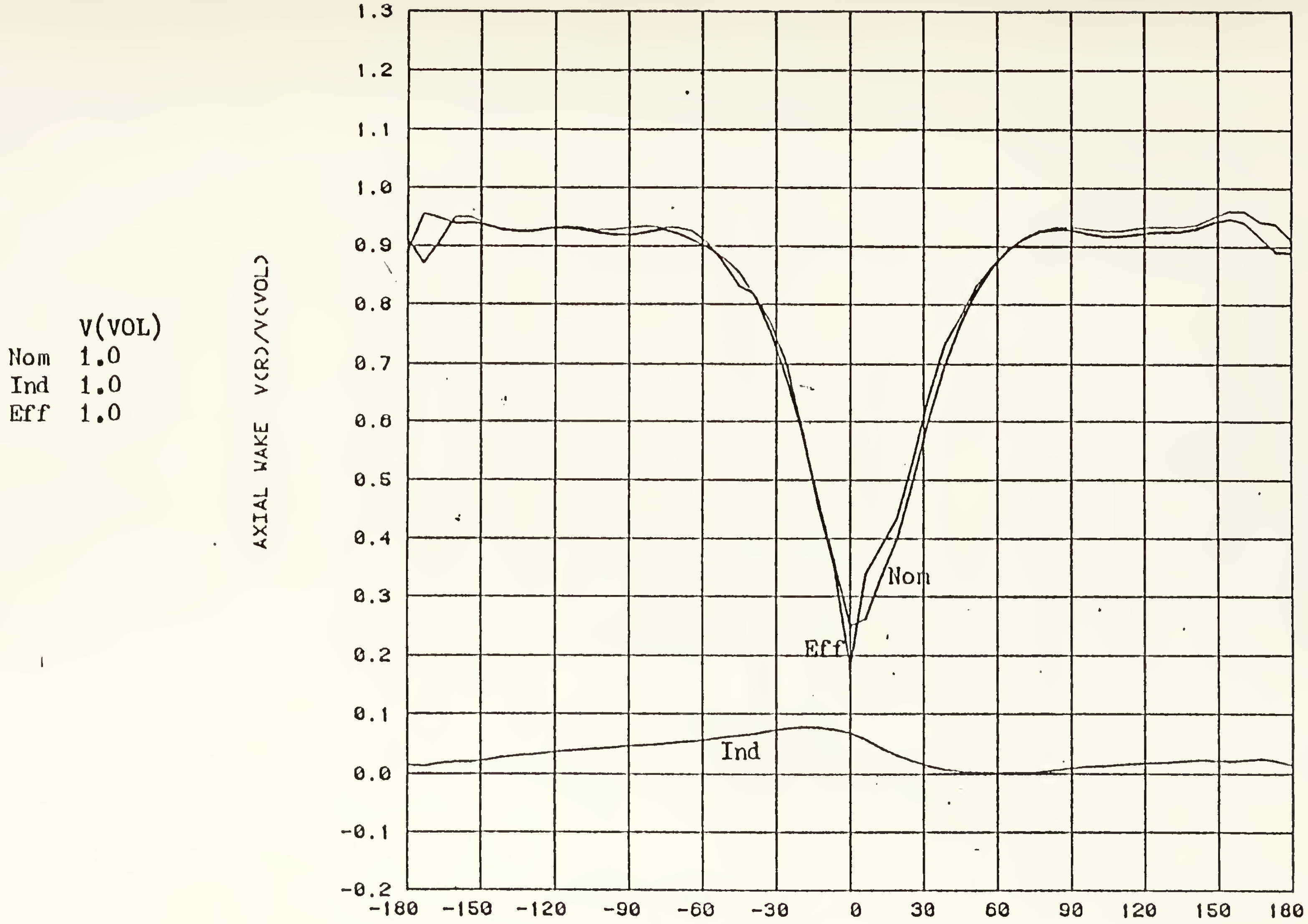


NOM, IND, EFF VELOCITIES, .80 RADIUS

	V(VOL)
Nom	1.0
Ind	1.0
Eff	1.0



NOM, IND, EFF VELOCITIES, .90 RADIUS.

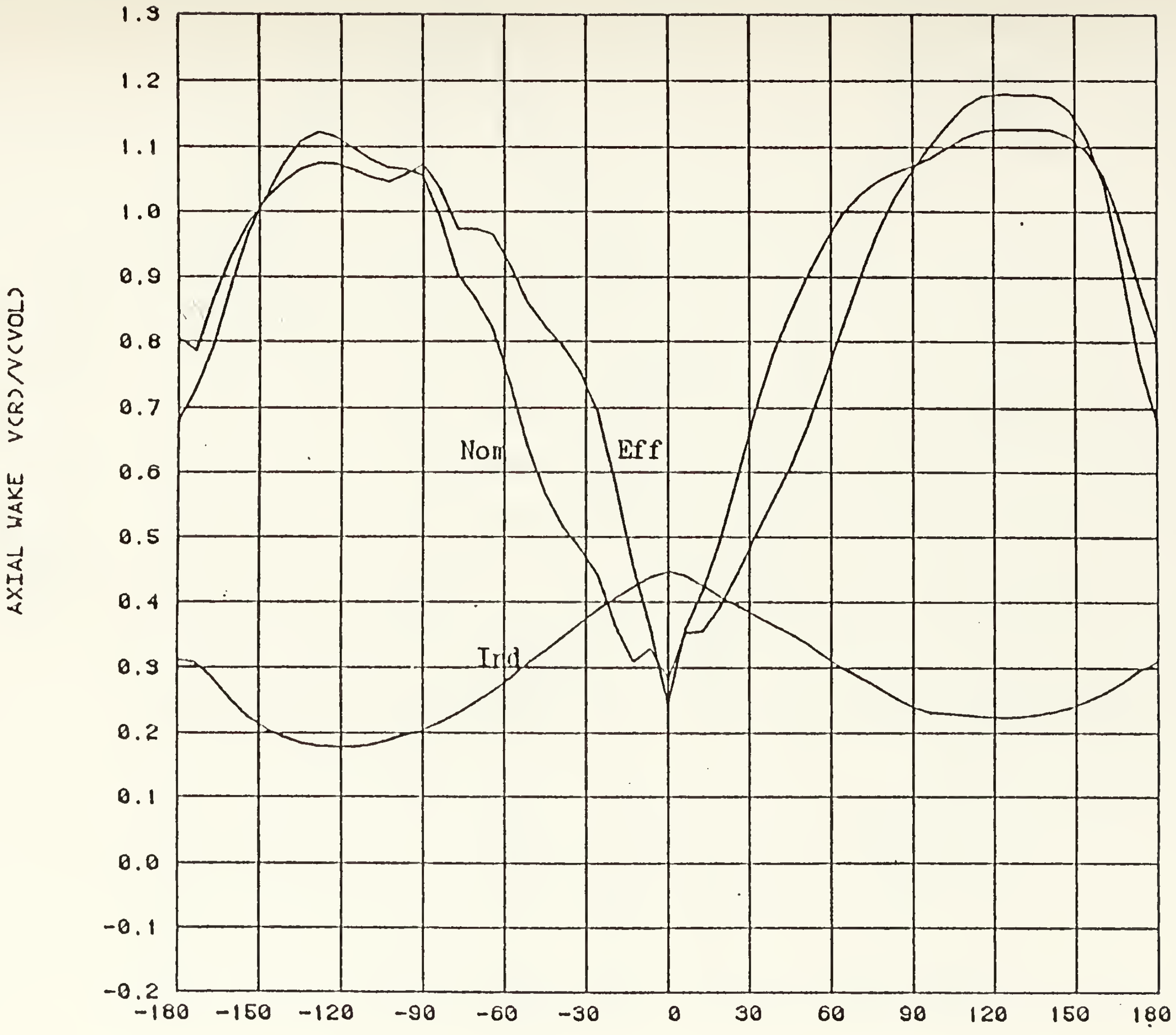


NOM, IND, EFF VELOCITIES, .98 RADIUS

** APPENDIX II **

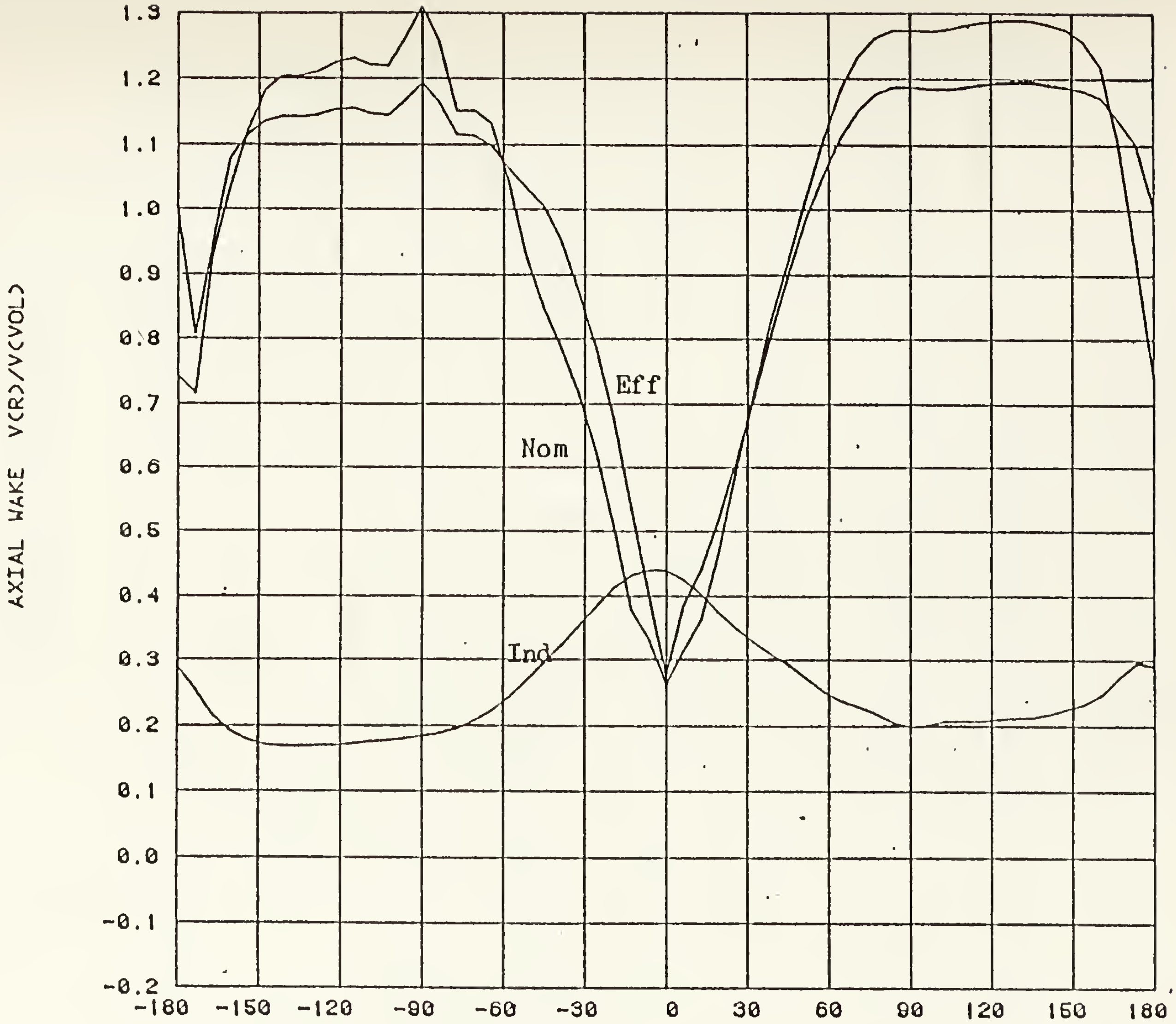
Nominal, Induced, and Effective Velocities, Non-Dimensionalized

	V(VOL)
Nom	0.732
Ind	1.0
Eff	0.787



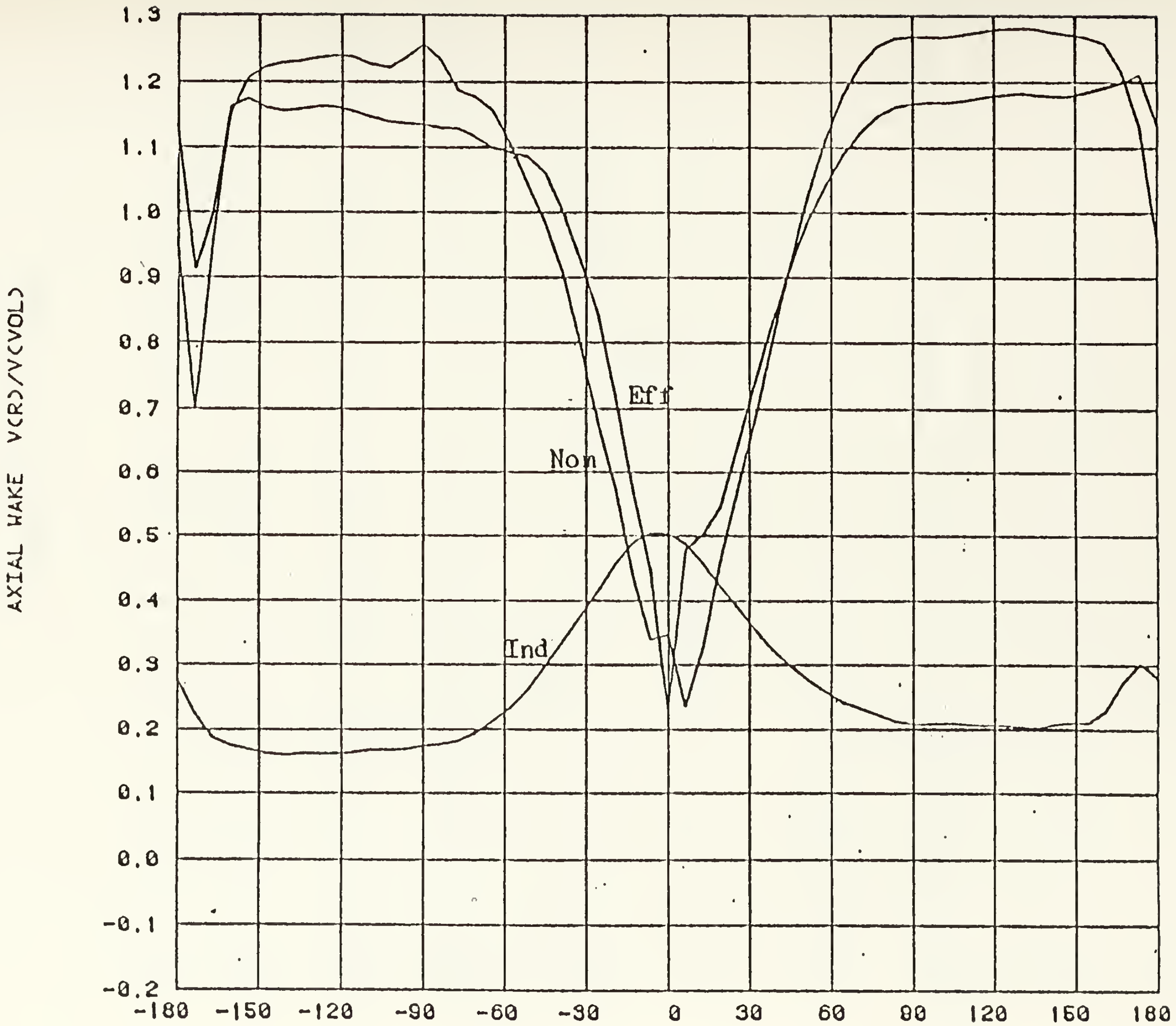
NOM, IND, EFF VELOCITIES, .40 RADIUS

	V(VOL)
Nom	0.732
Ind	1.0
Eff	0.787



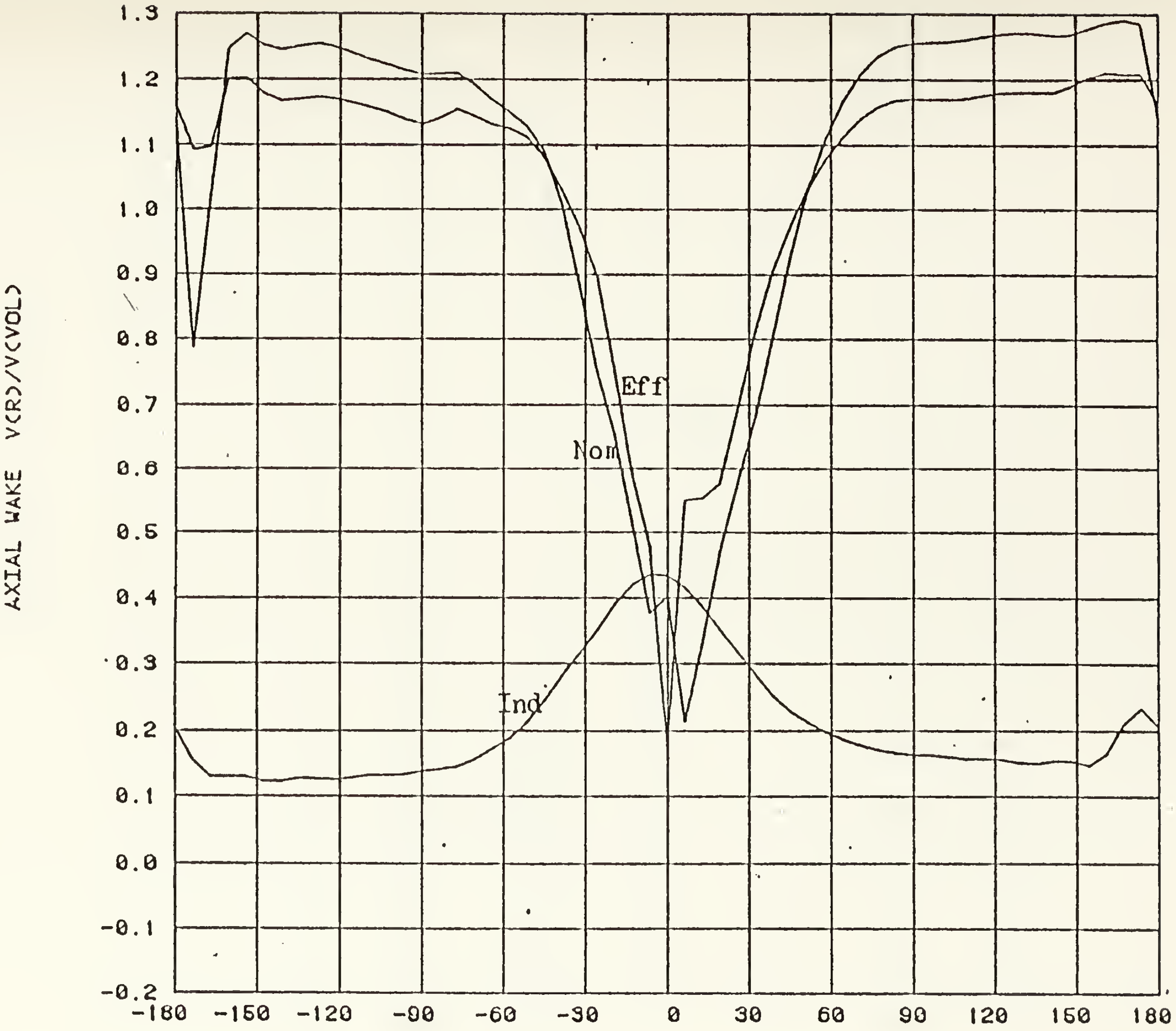
NOM, IND, EFF VELOCITIES, .60 RADIUS

	V(VOL)
Nom	0.732
Ind	1.0
Eff	0.787



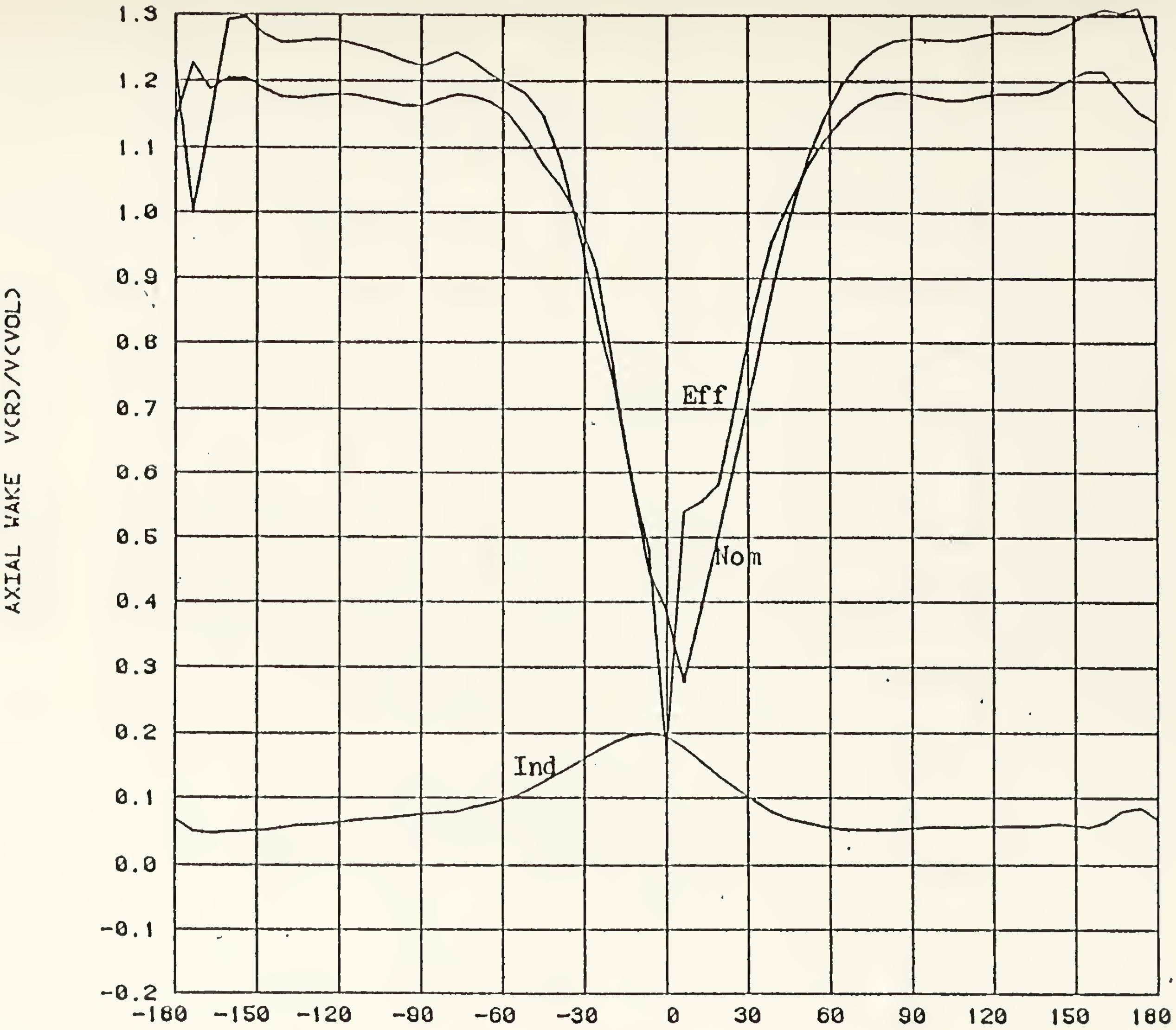
NOM, IND, EFF VELOCITIES, .70 RADIUS

	V(VOL)
Nom	0.732
Ind	1.0
Eff	0.787



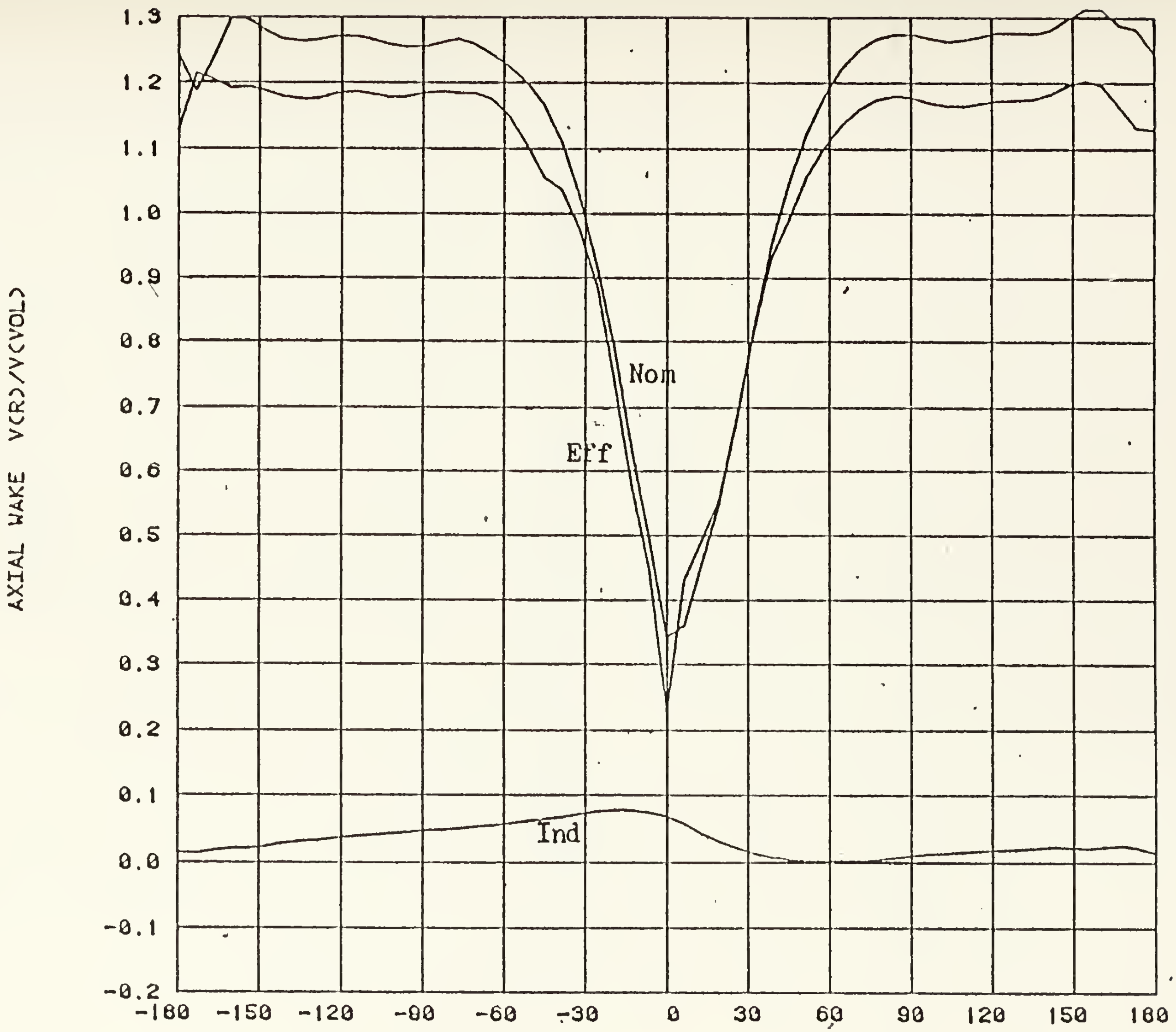
NOM, IND, EFF VELOCITIES, .80 RADIUS

	V(VOL)
Nom	0.732
Ind	1.0
Eff	0.787



NOM, IND, EFF VELOCITIES, .90 RADIUS

	V(VOL)
Nom	0.732
Ind	1.0
Eff	0.787

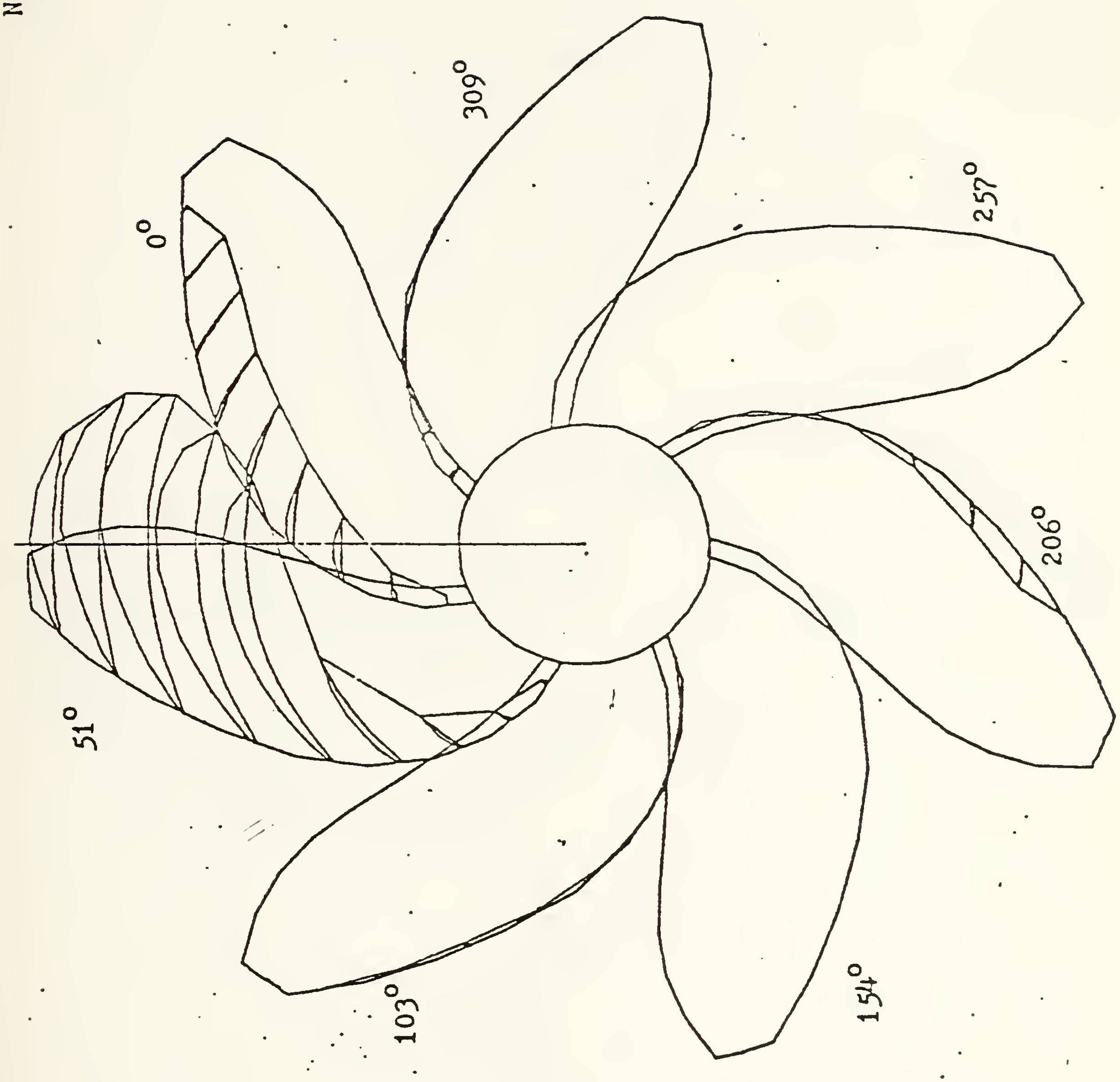


NOM, IND, EFF VELOCITIES, .98 RADIUS

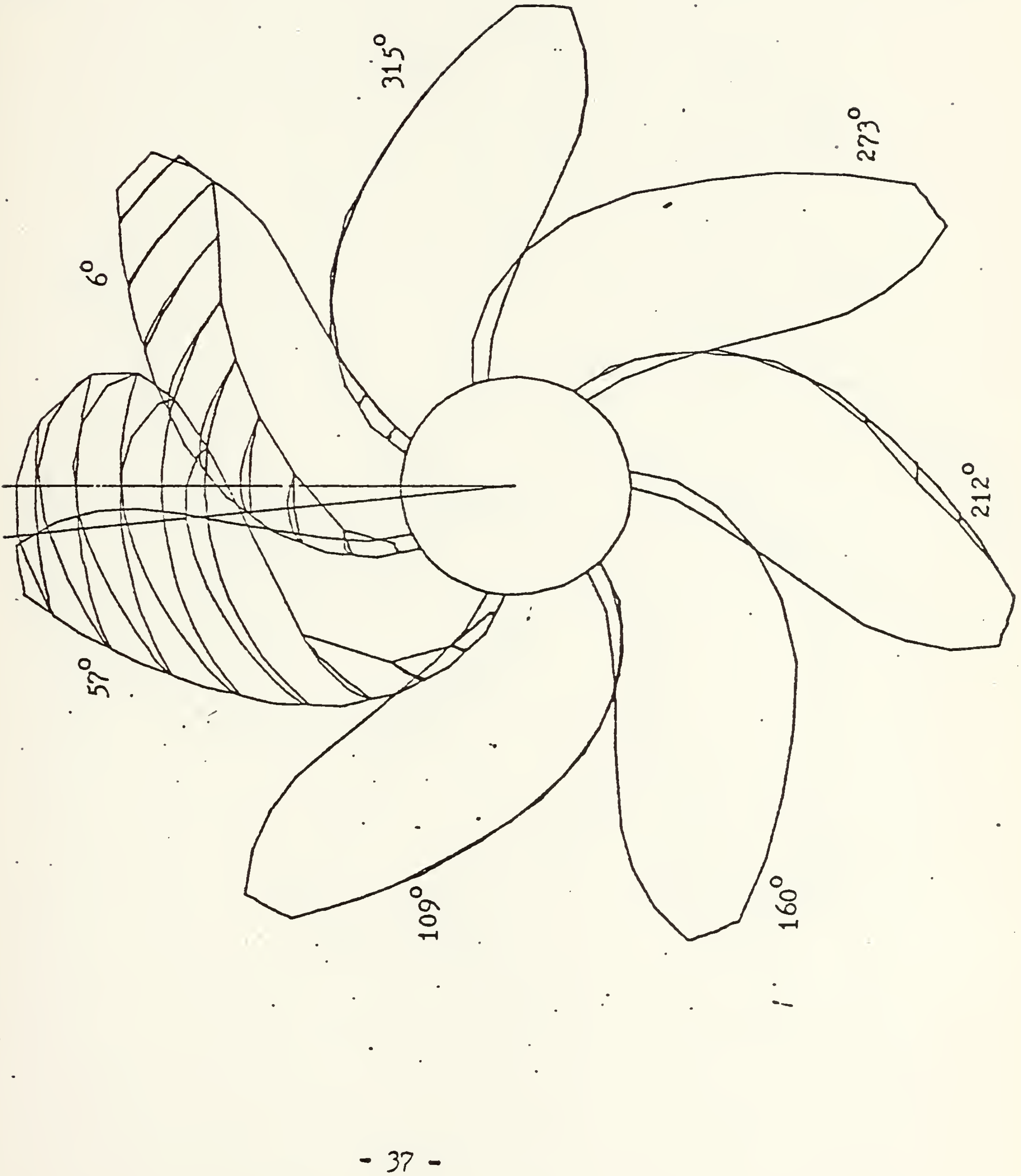
** APPENDIX III **

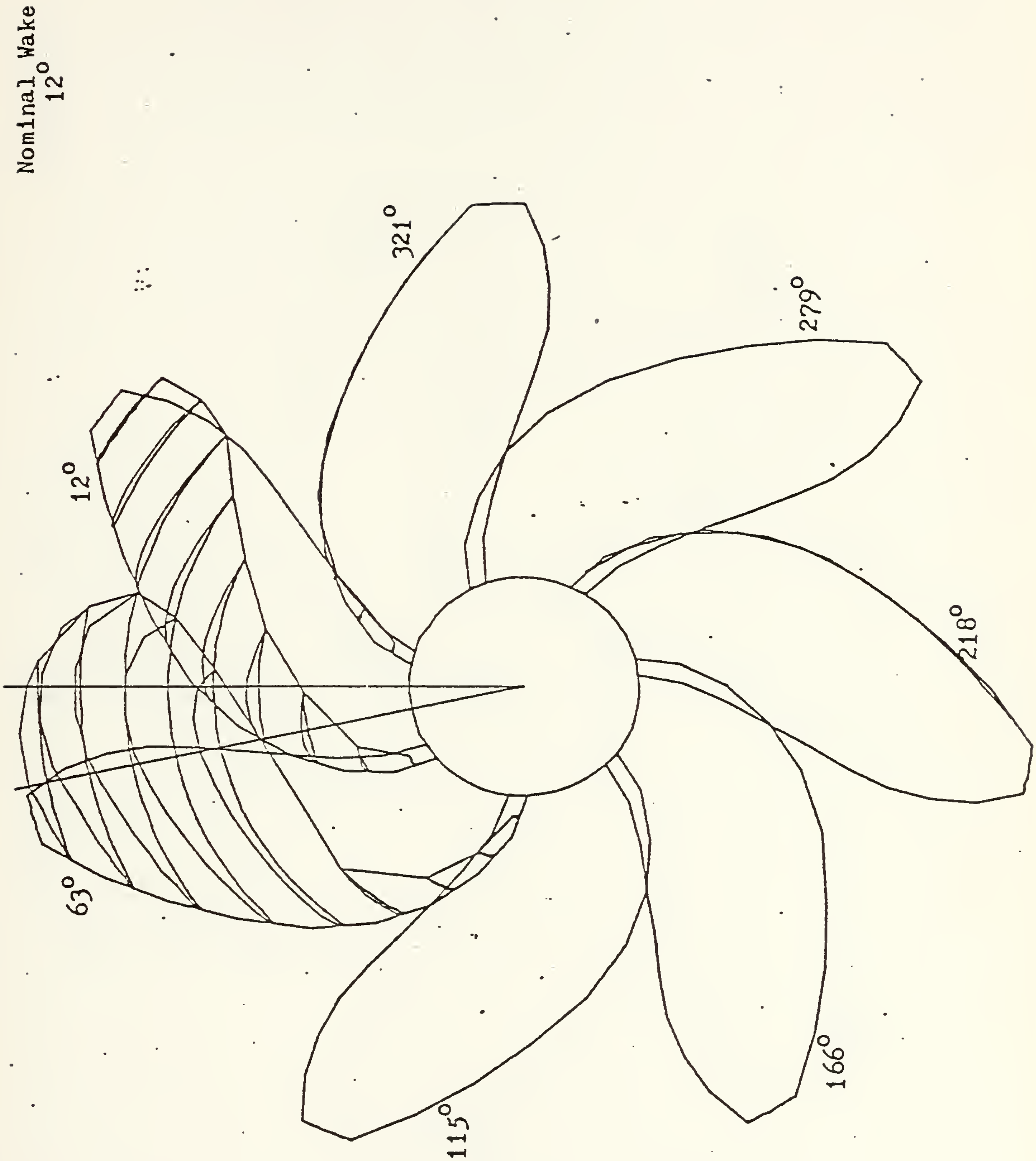
Propeller Cavitation, Nominal Wake

Nominal Wake
.0°

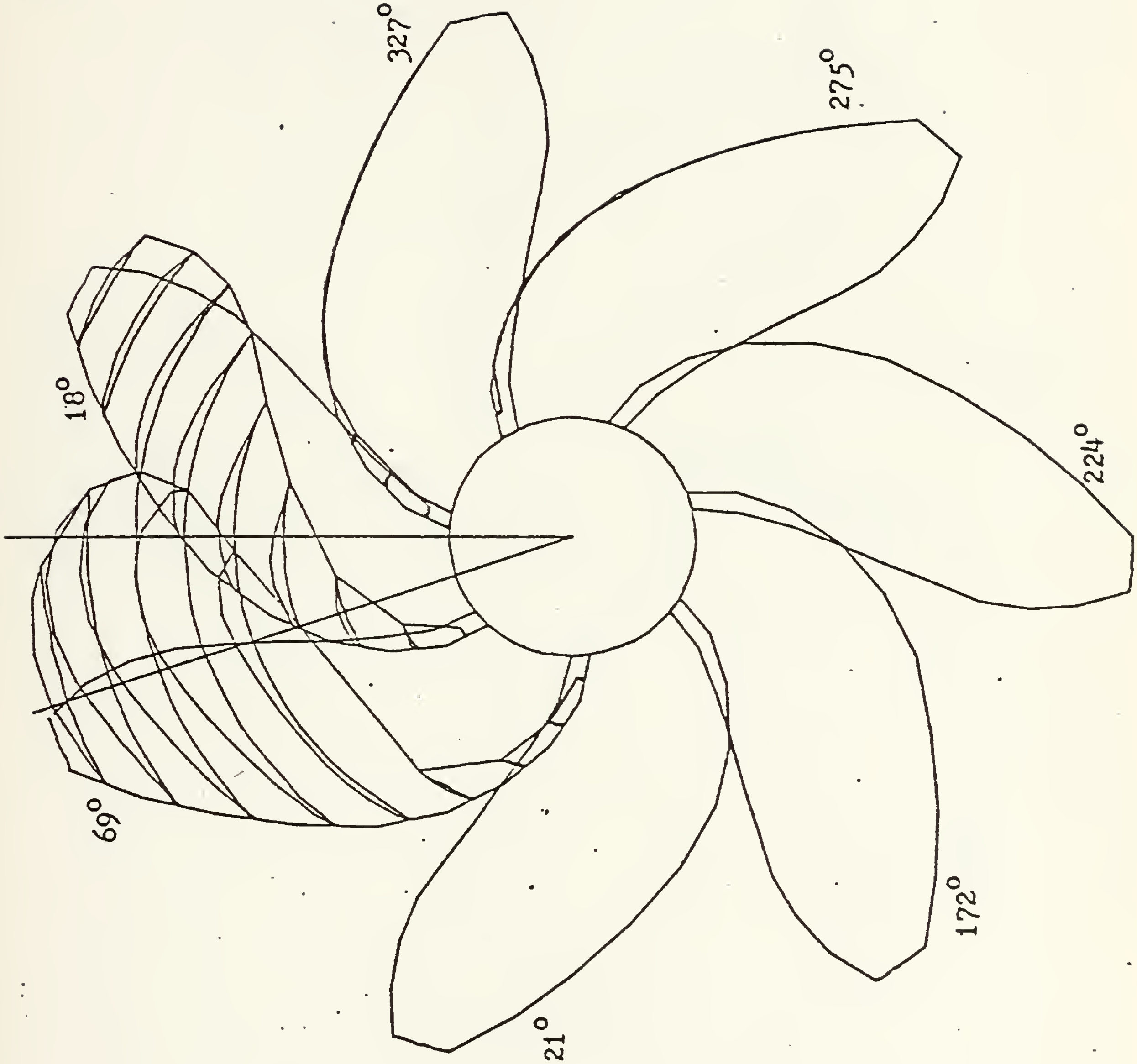


Nominal Wake 6°

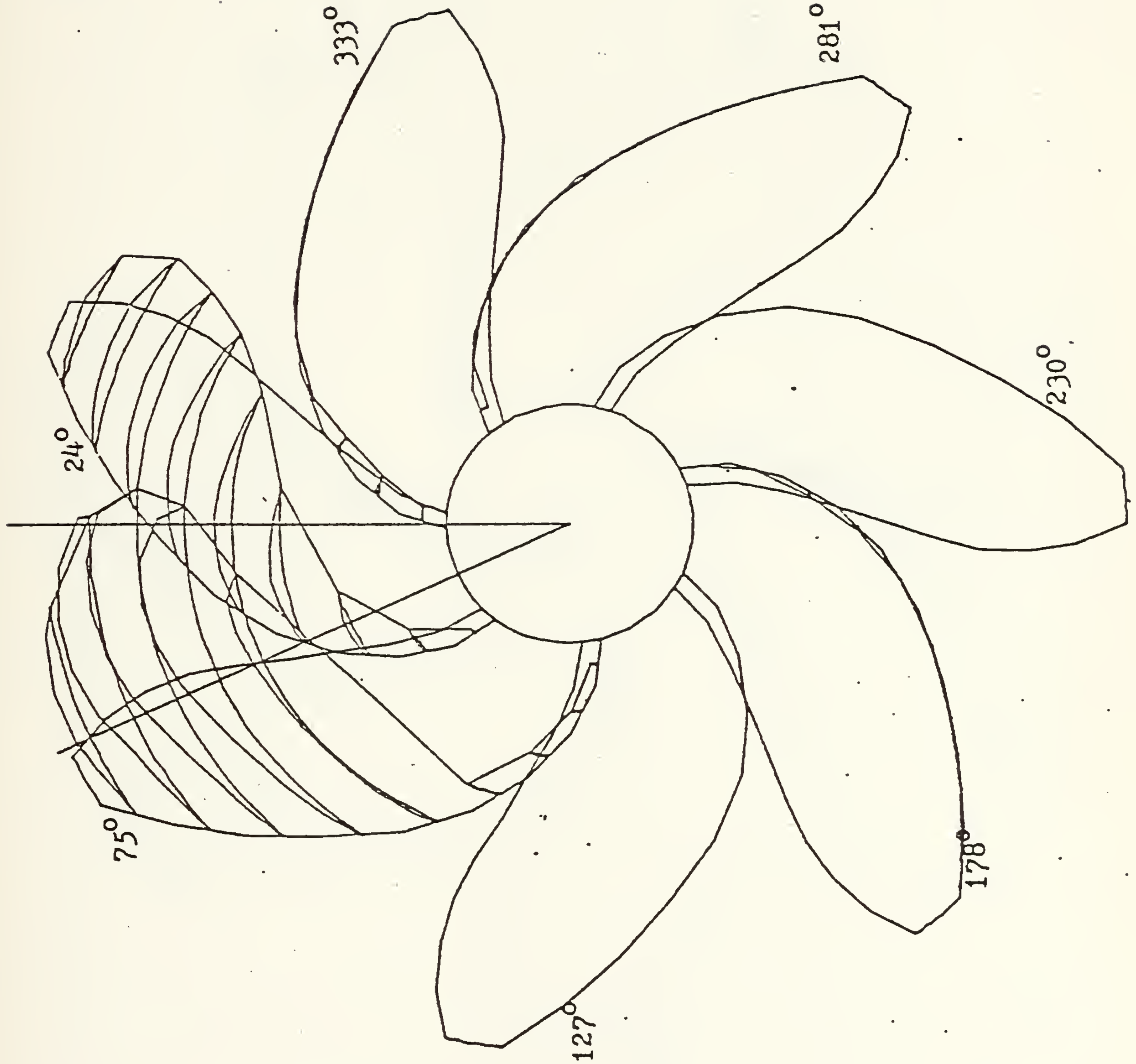


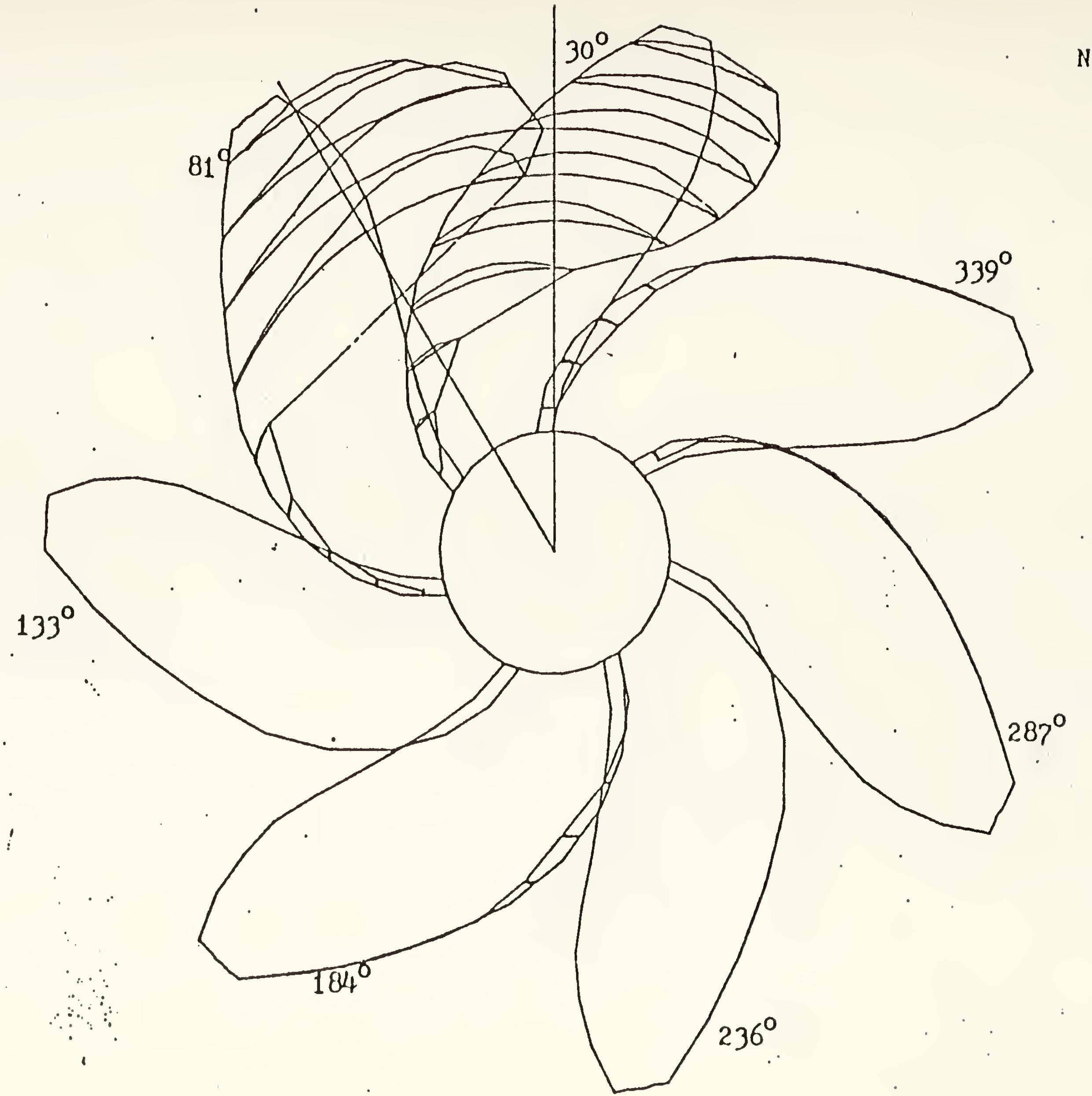

Nominal Wake
12°
321°
279°
12°
218°
63°
166°
115°

Nominal Wake
18°



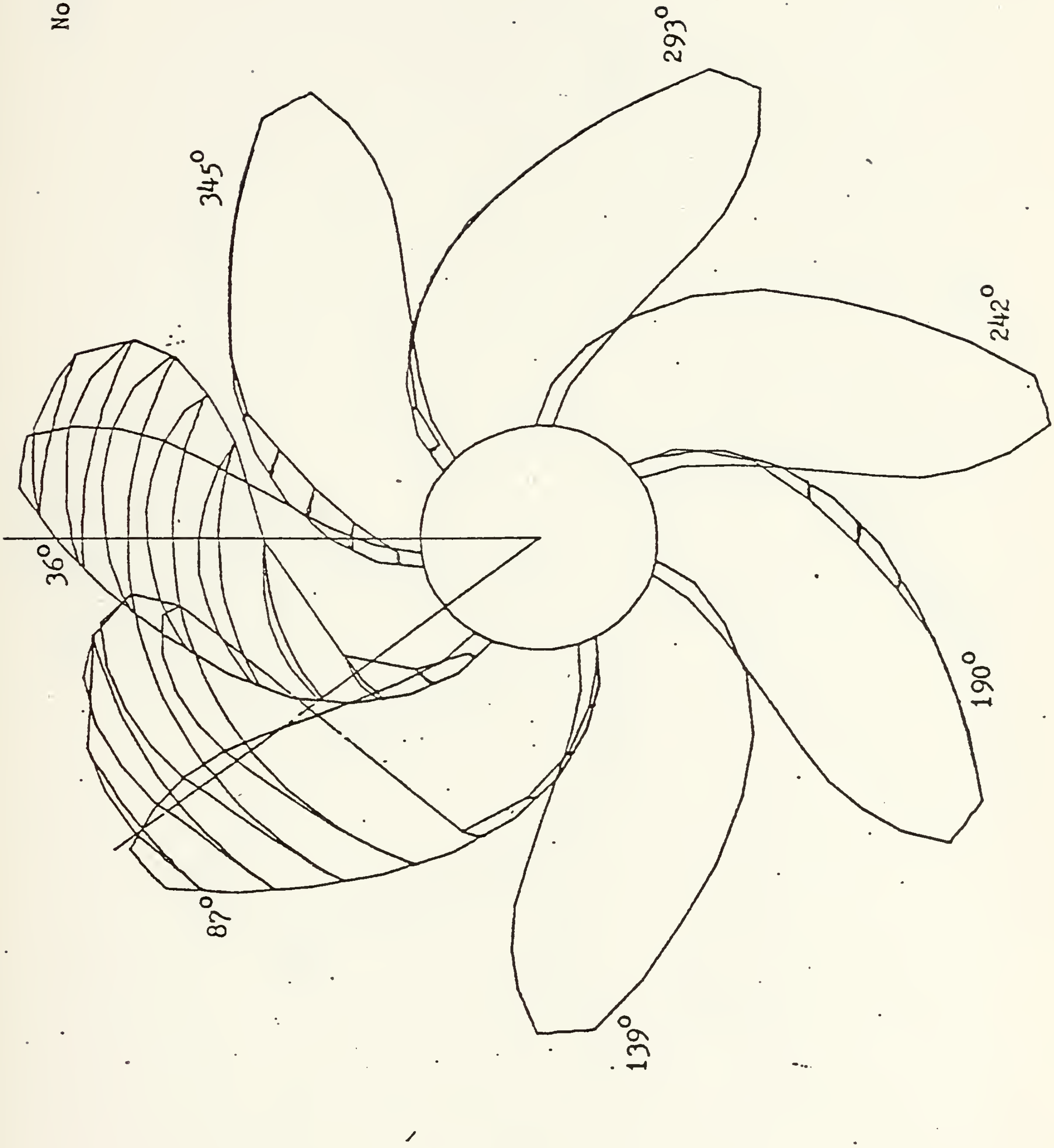
Nominal Wake
24°



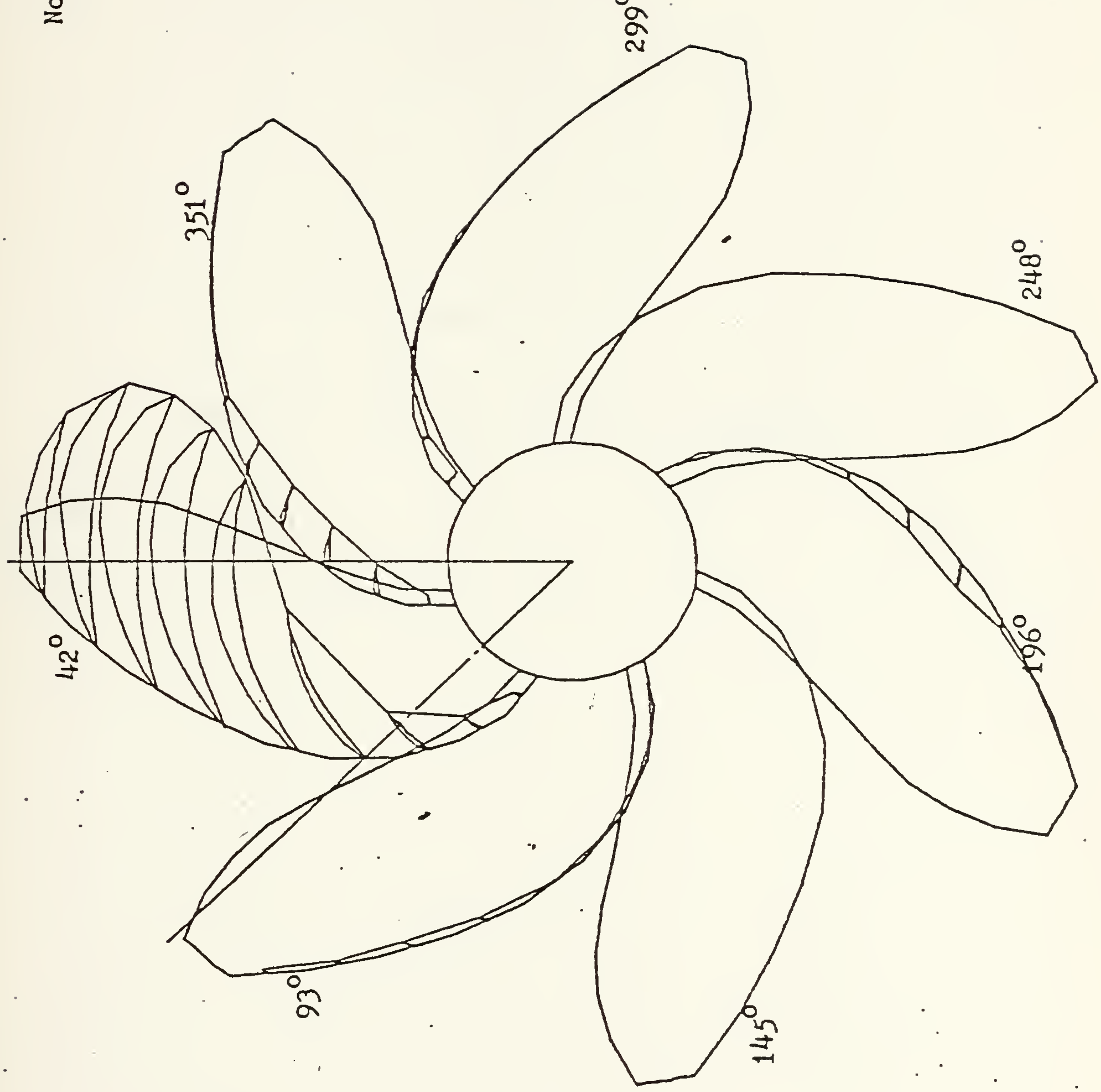


Nominal Wake
30°

Nominal Wake
36°

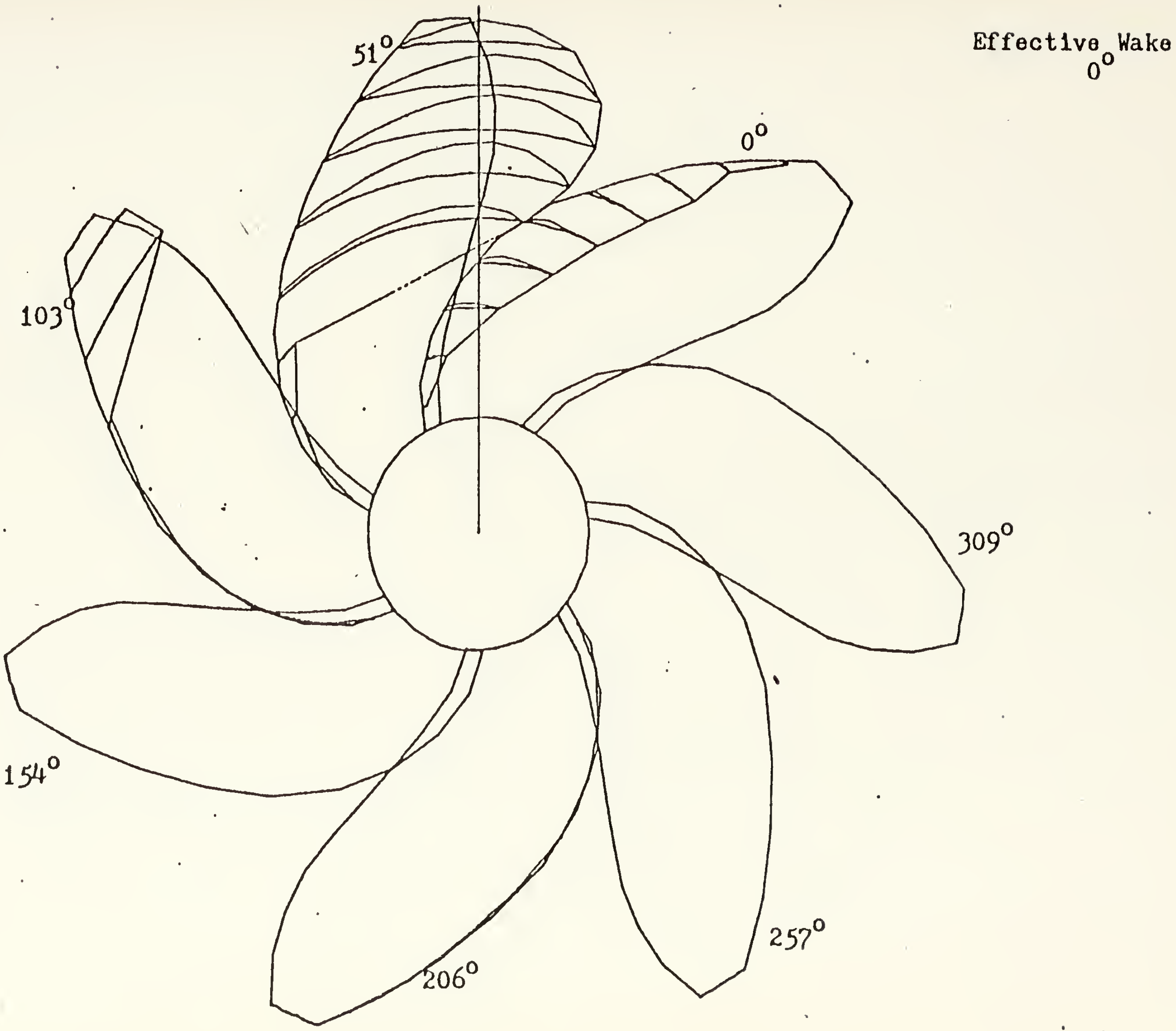


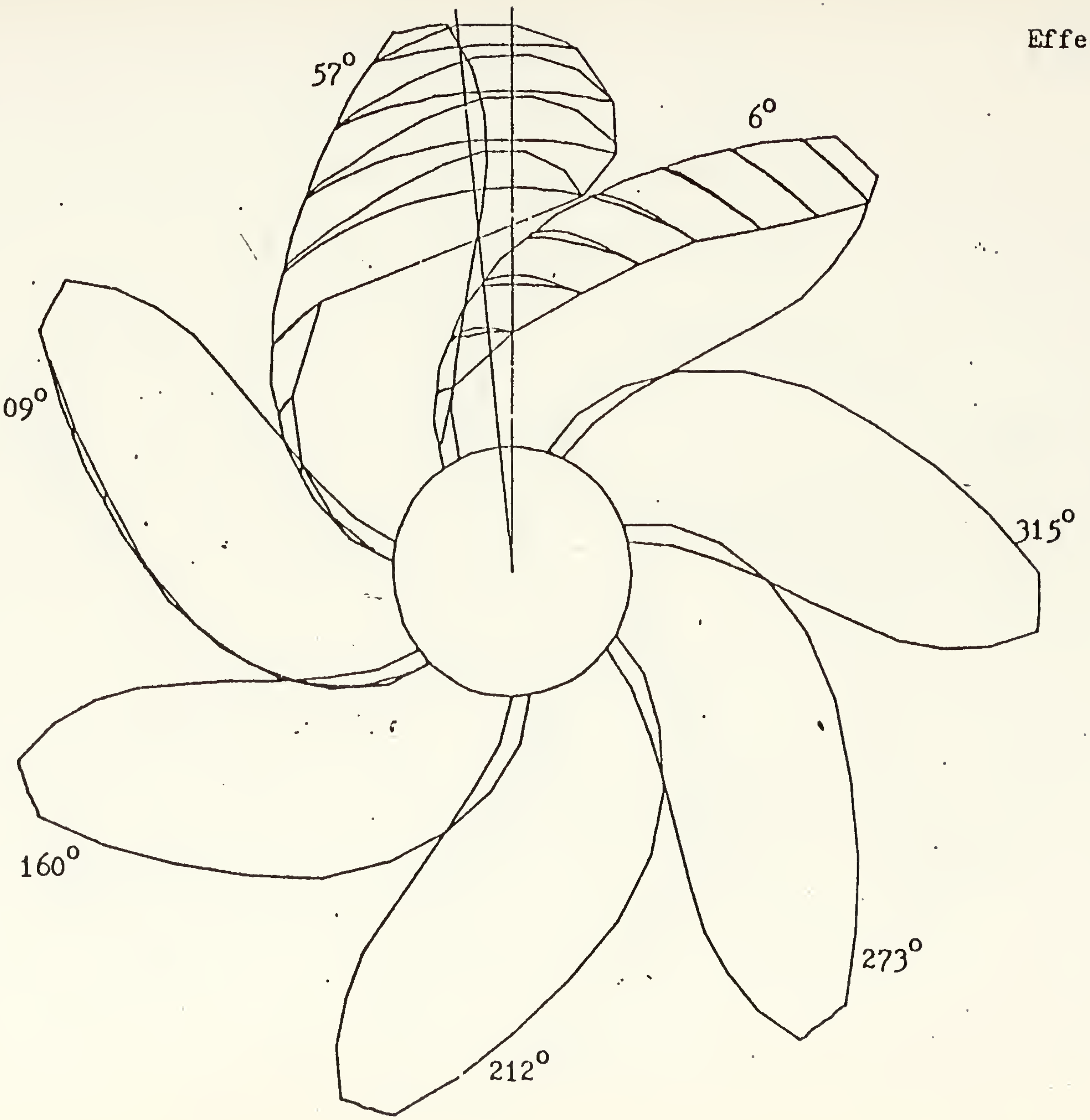
Nominal Wake
42°



** APPENDIX IV **

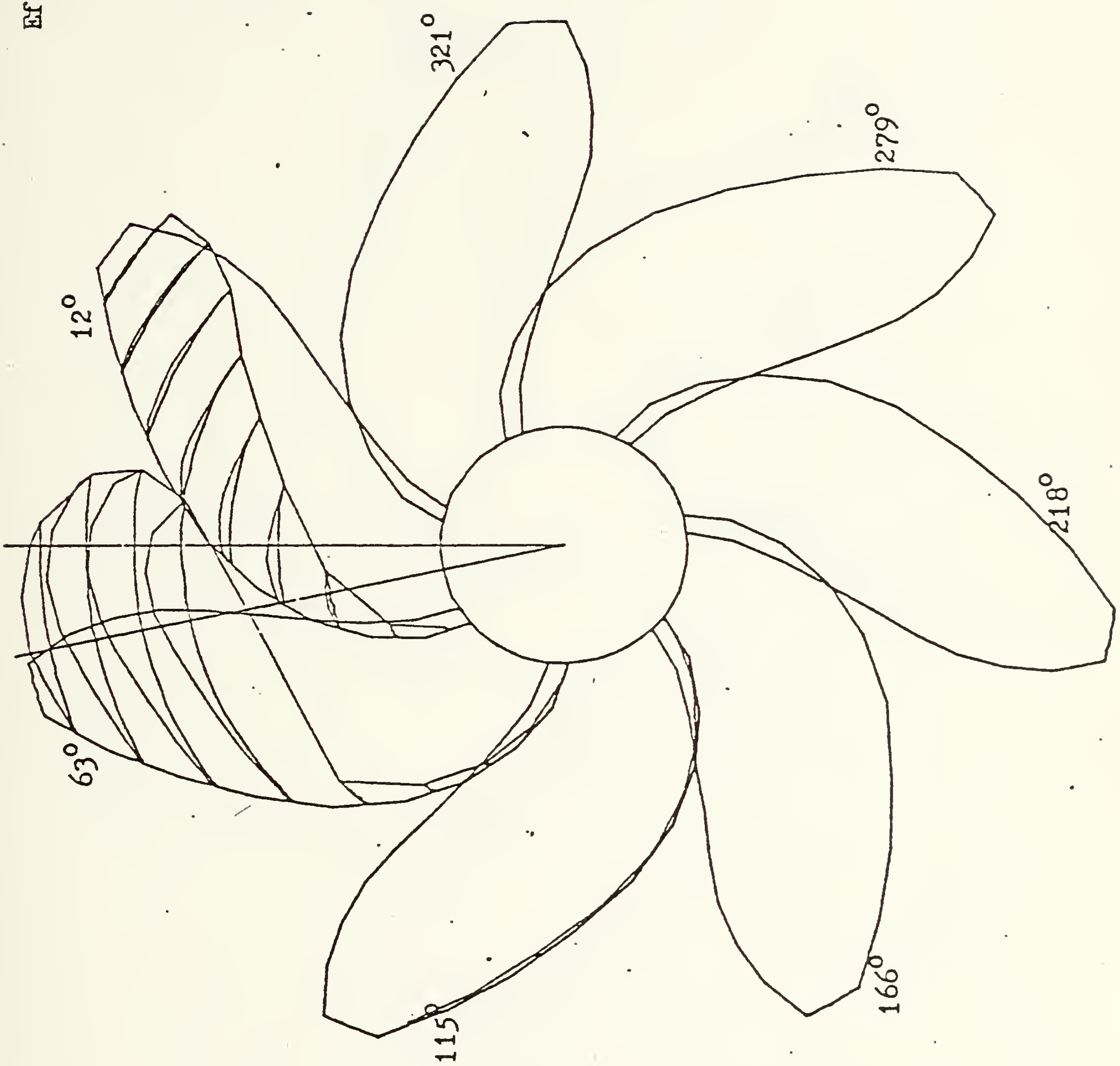
Propeller Cavitation, Effective Wake


Effective Wake
0°
51°
0°
103°
309°
154°
257°
206°

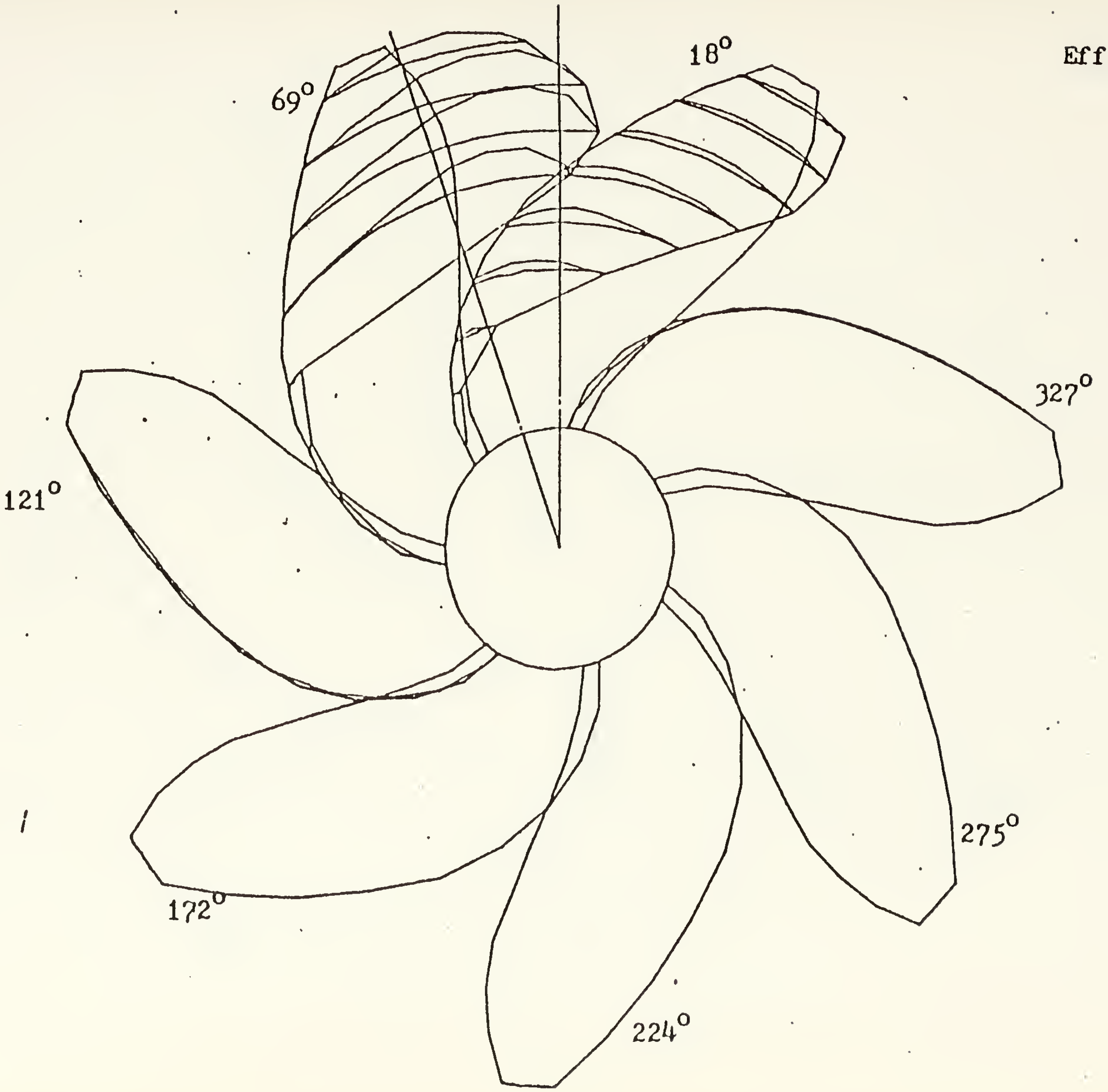


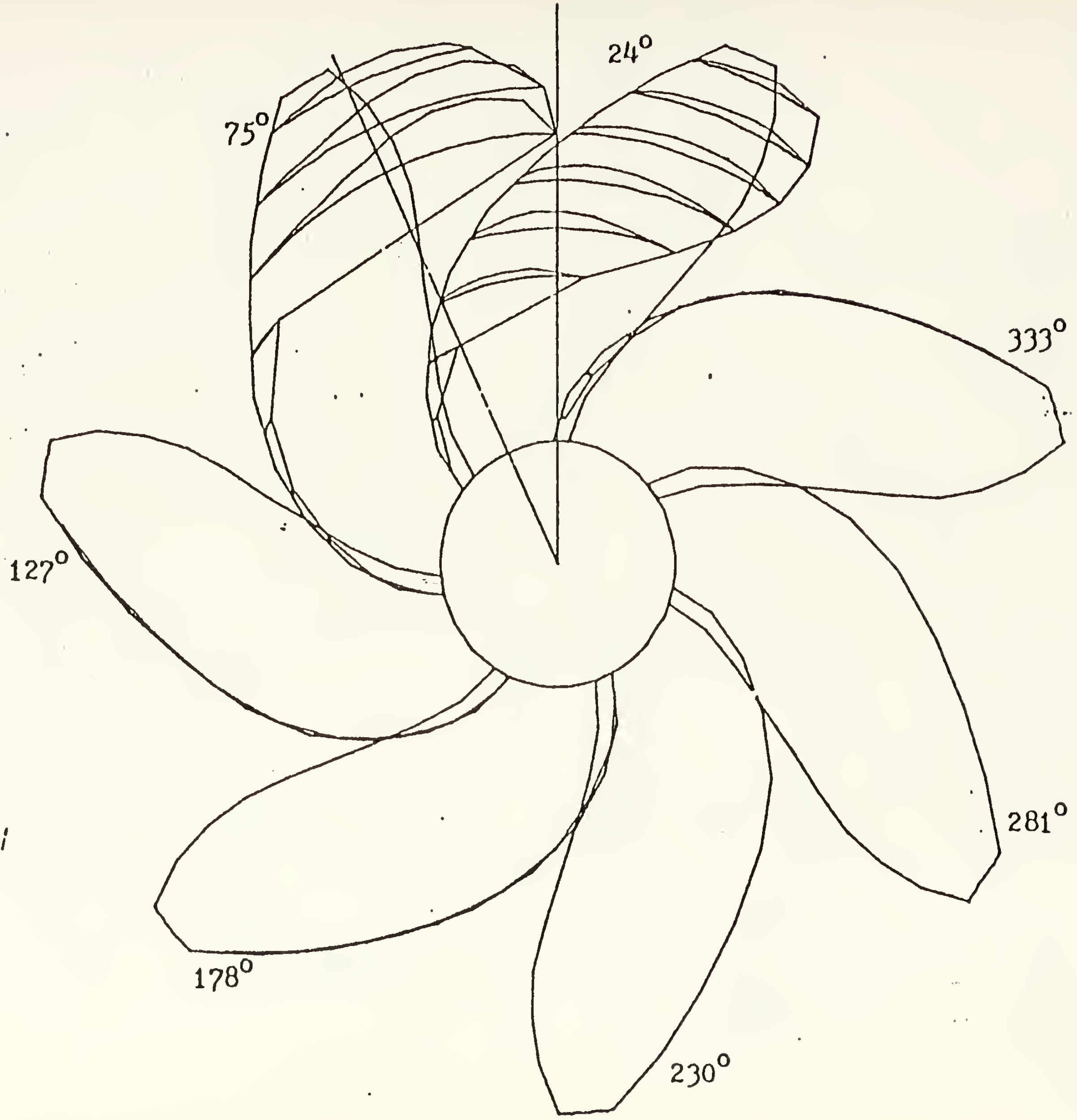
Effective Wake
6°

Effective Wake
12°



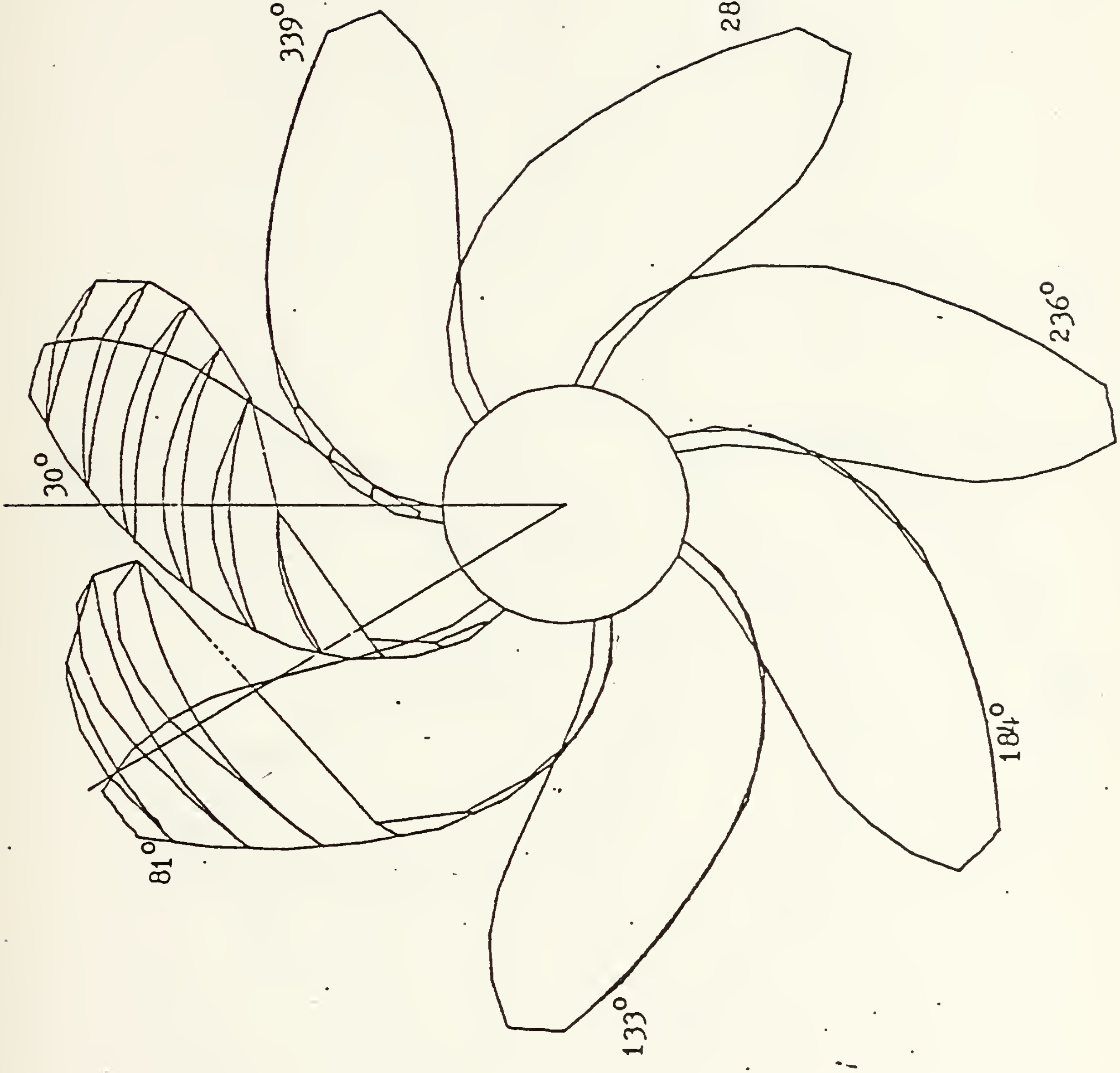
Effective Wake
18°



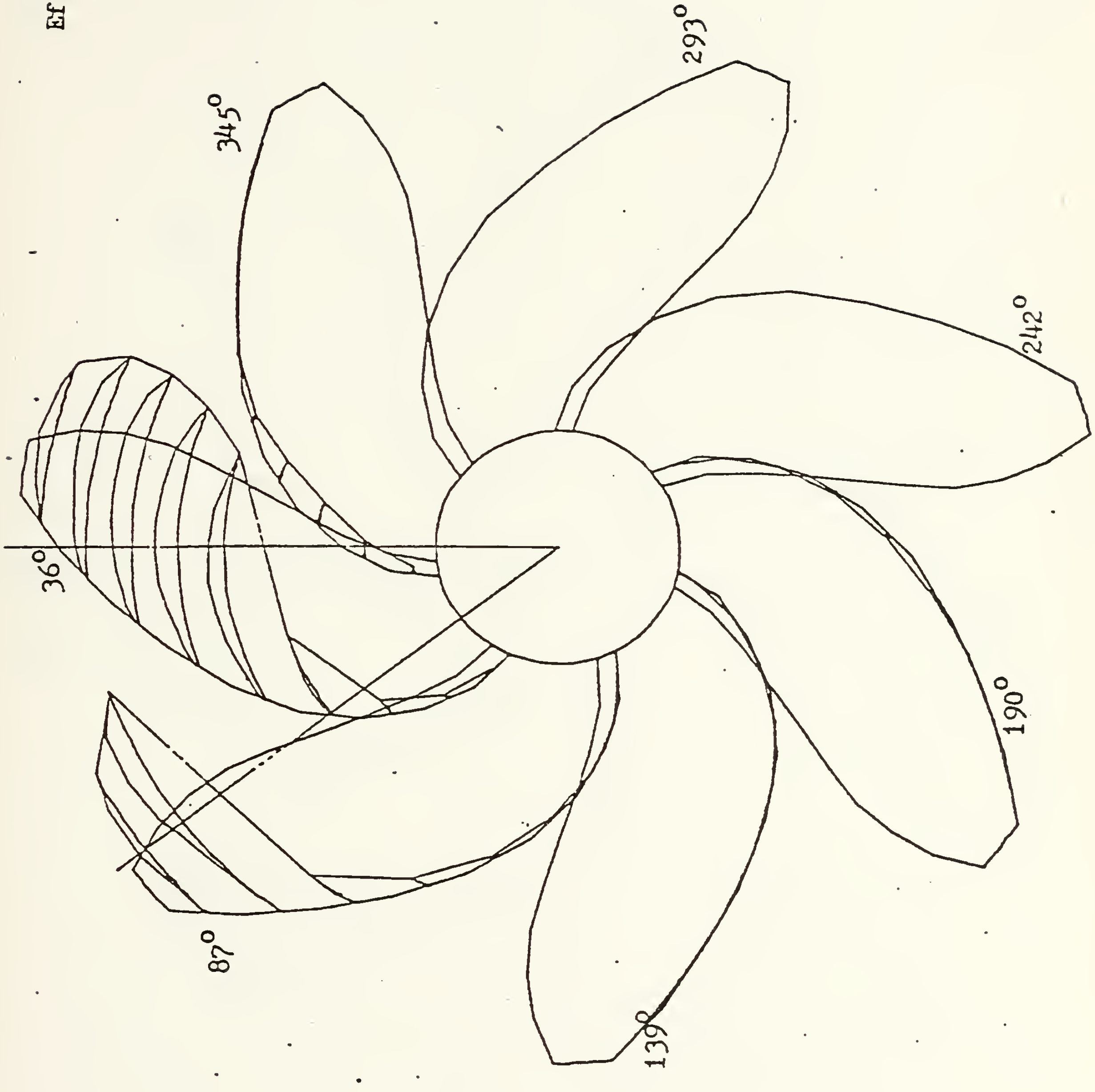


Effective Wake
24°

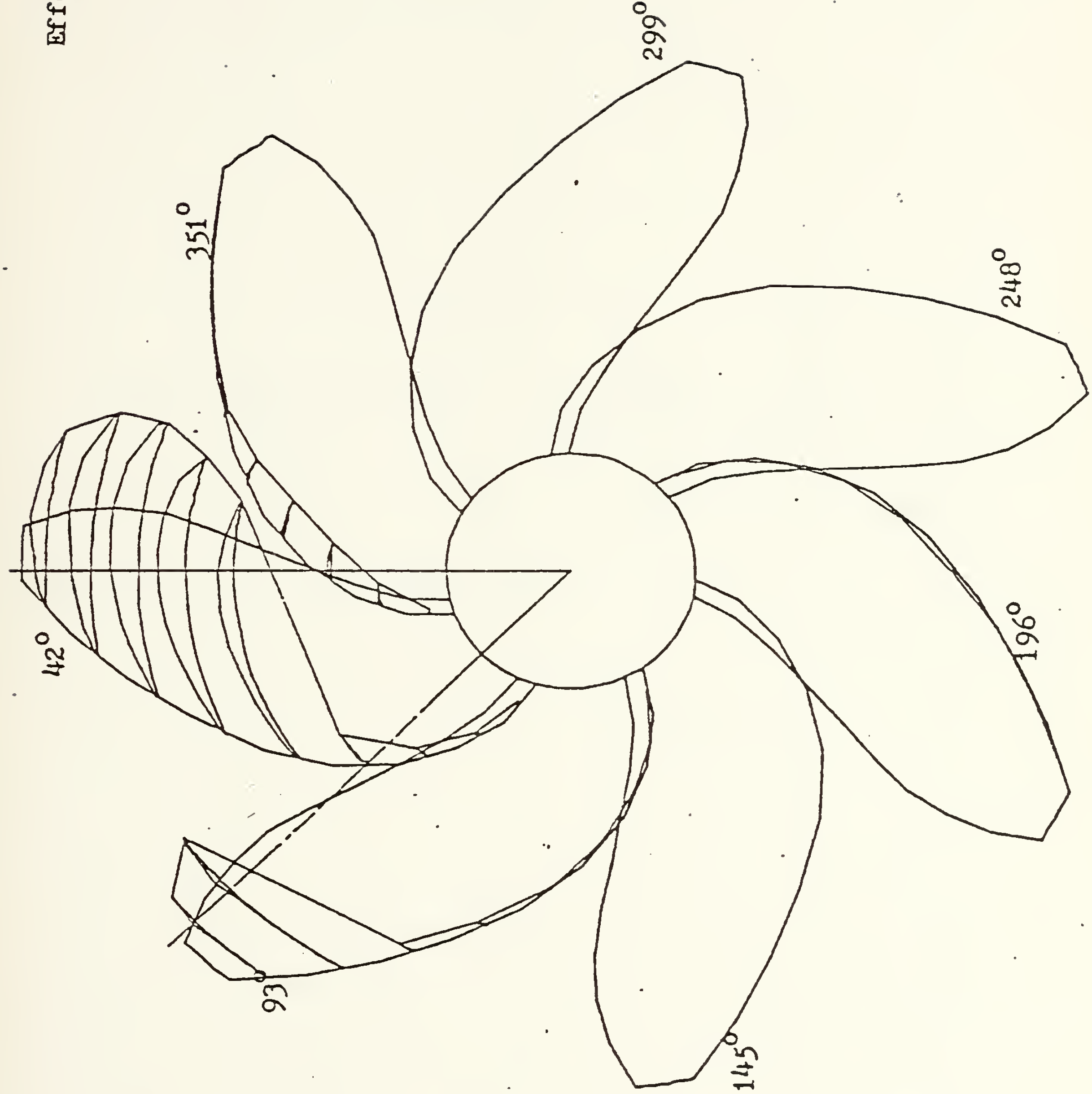
Effective Wake
30°



Effective Wake
36°



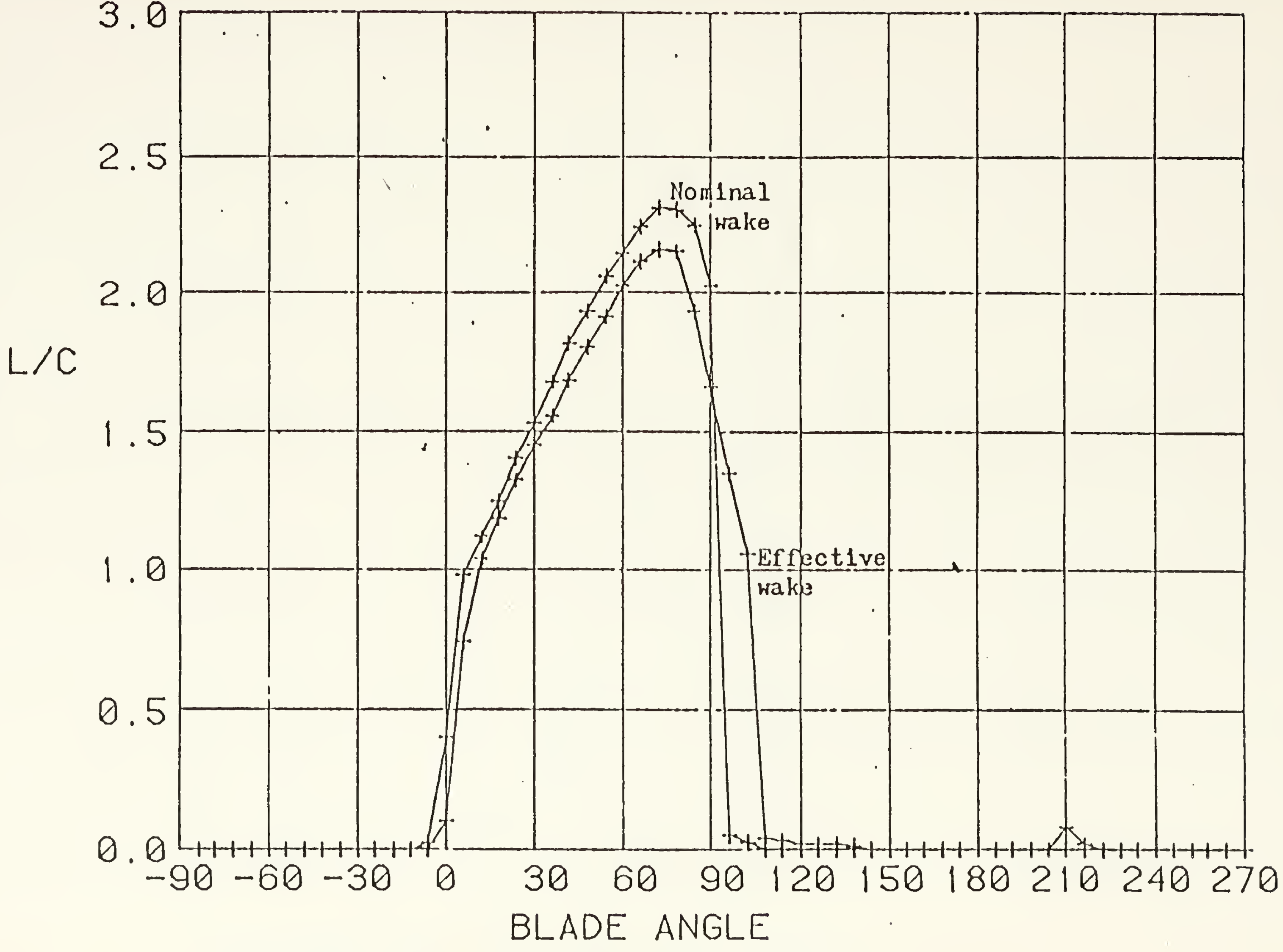
Effective Wake
42°



** APPENDIX V **

Cavity Length, Nominal and Effective Wakes

PUF-3 CAVITY LENGTH OUTPUT

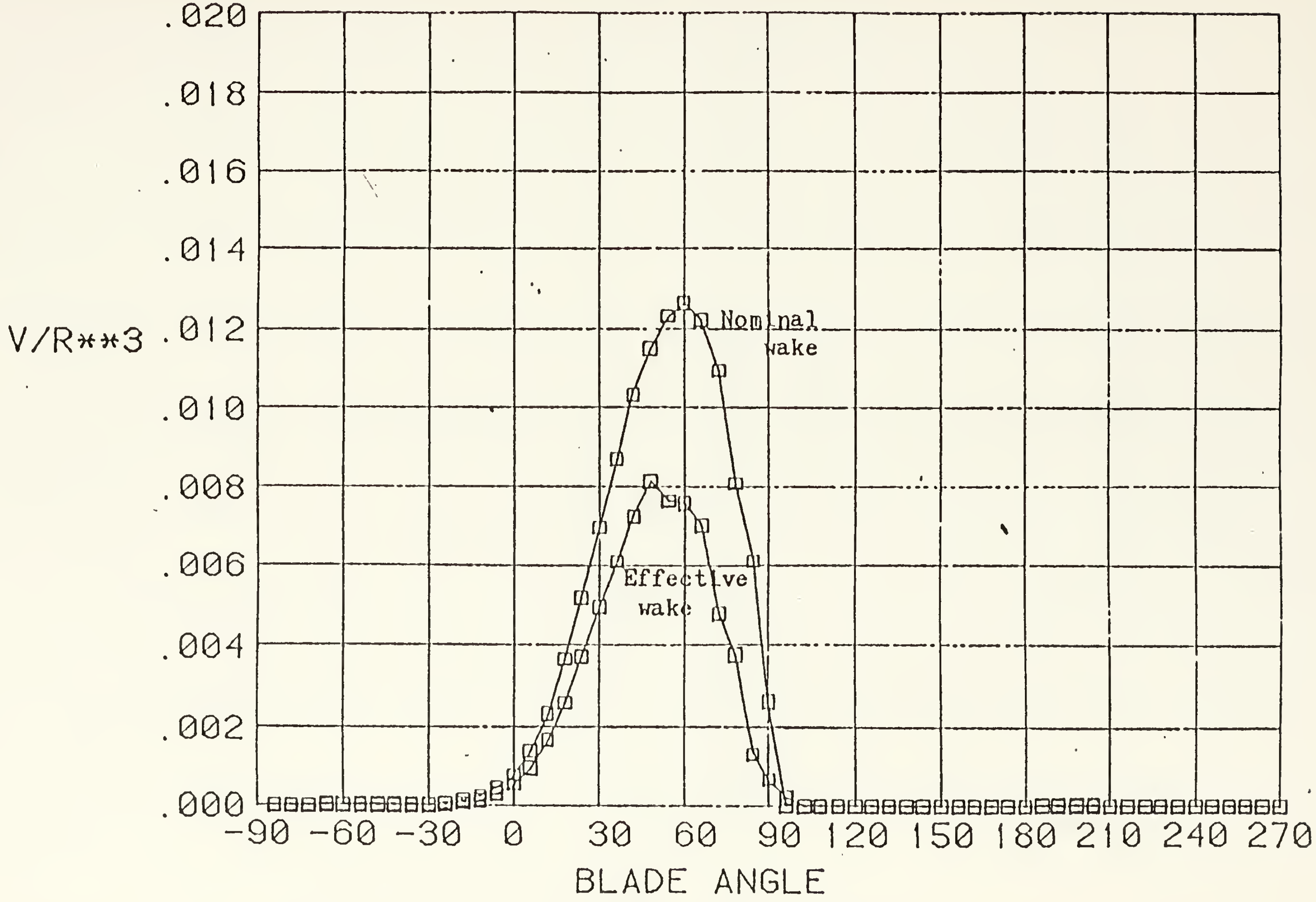


NOMINAL VS. EFFECTIVE WAKE

** APENDIX VI **

Cavity Volume, Nominal and Effective Wakes

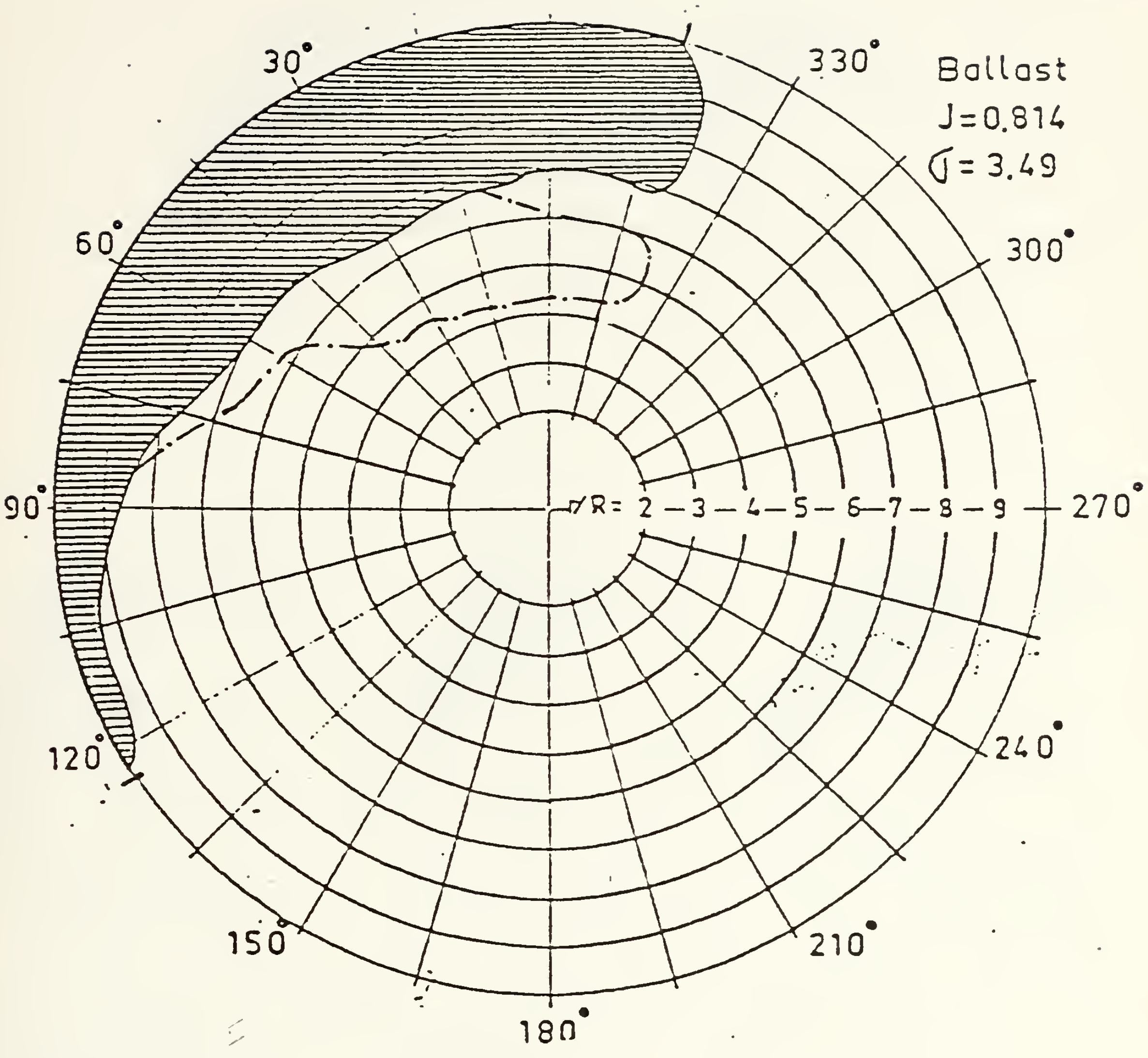
PUF-3 CAVITY VOLUME OUTPUT



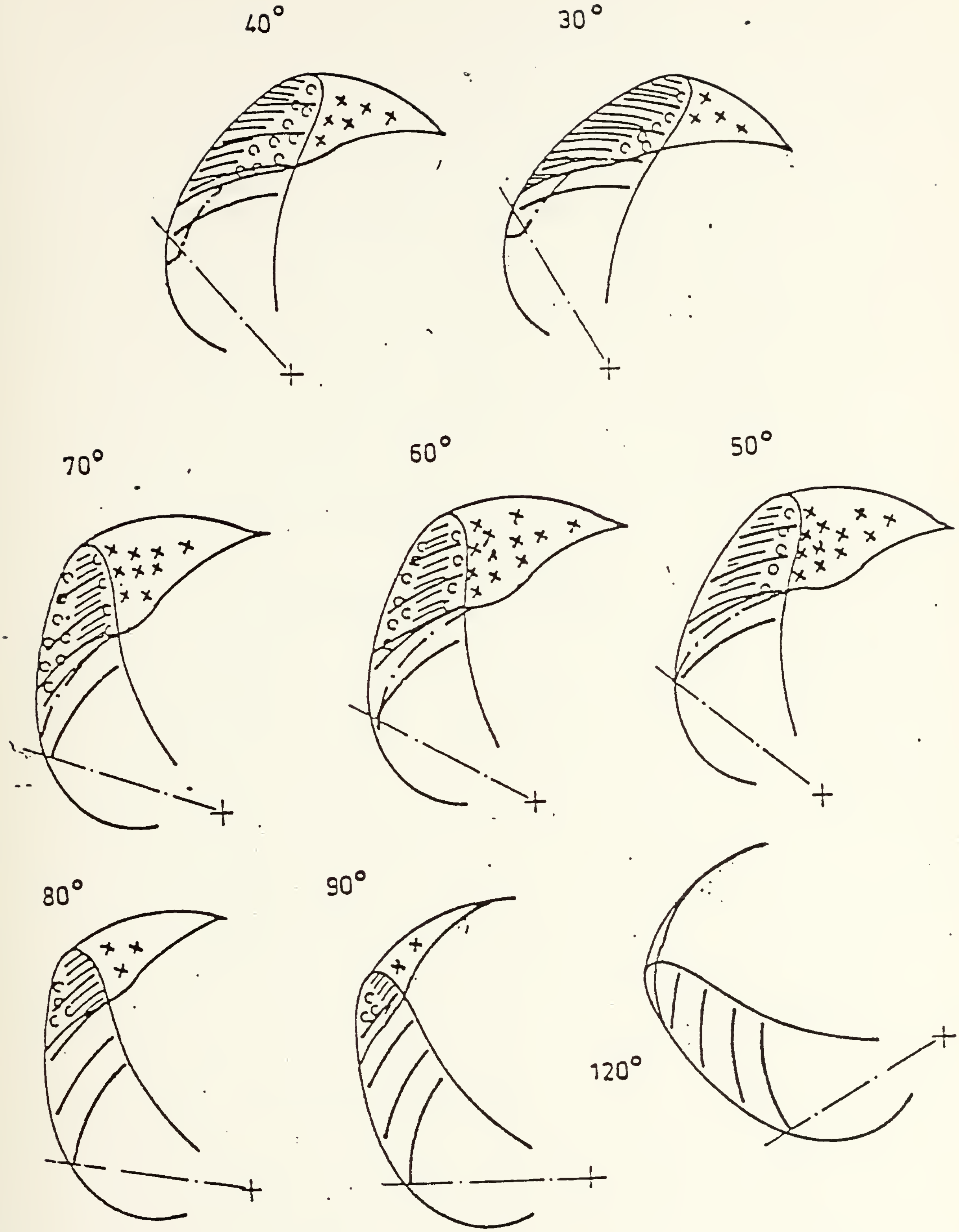
NOMINAL VS. EFFECTIVE WAKE

** APPENDIX VII **

SSPA Experimental Propeller Cavitation Extent Plots



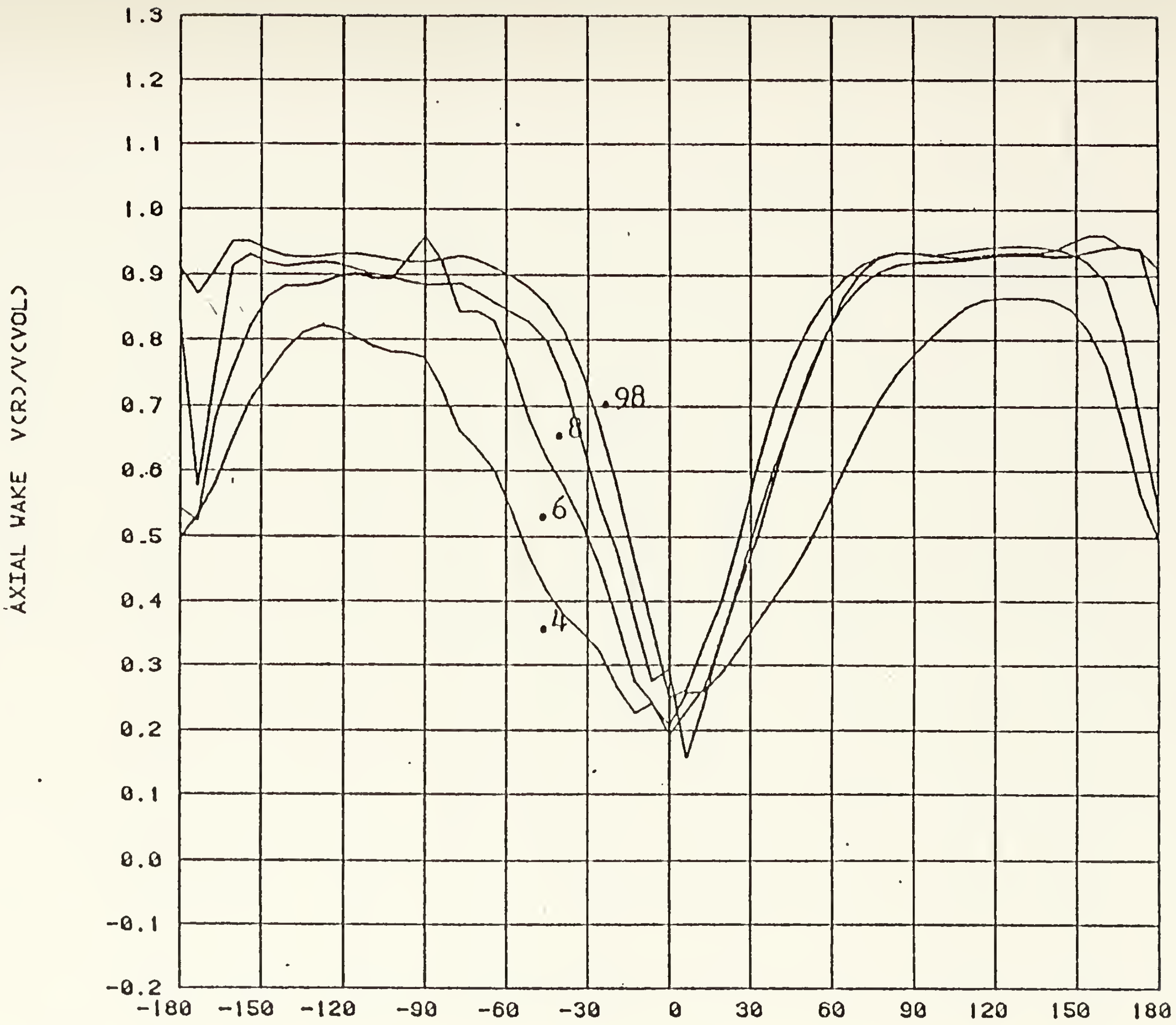
Reference (10)



Reference (10)

** APPENDIX VIII **

Nominal Wake Diagram

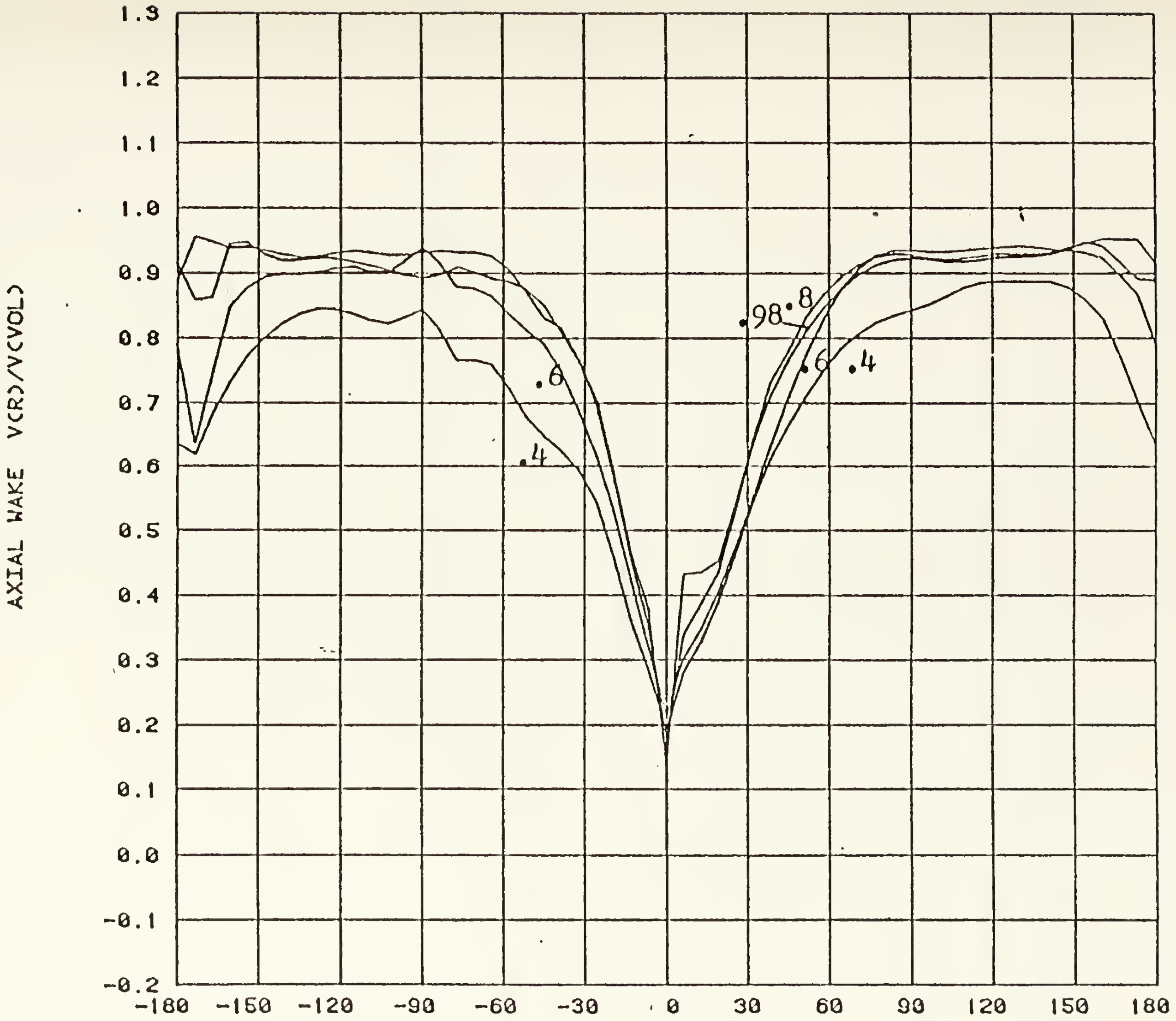


Actual Velocities (I.e., V(VOL)=1.0)

NOMINAL WAKE, .4, .6, .8, .98 RADII

** APPENDIX IX

Effective Wake Diagram



Actual Velocities (I.e., V(VOL)=1.0)

EFFECTIVE WAKE, .4, .6, .8, .98 RADII

** APPENDIX X **

UFPV Program Listing

```
C     ******************************************************
C     *                                                    *
C     *   MAIN PROGRAM IN FPUV, A PROGRAM TO CALCULATE     *
C     *   UNSTEAOY VELOCITY HARMONICS AT FIELO POINTS      *
C     *   SPECIFIEO INTERACTIVELY IN ROTERM.               *
C     *                                                    *
C     ******************************************************
C
C
C *** COMMON TO CONTAIN GEOMETRIC PARAMETERS
C
      COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH
C
C *** COMMON TO CONTAIN SINGULARITY STRENGTHS
C
      COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV
     & .,GAM(9,60),GMEAN(9),OKFW(20),HBWX,HBWY,HBWZ
C
C *** COMMON TO CONTAIN FIELO POINT OATA, NBLAOE, IOENT, NSR, ANO VSR
C
      COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,
     &   OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPLOT,OIAM
C
C *** COMMON TO CONTAIN VELOCITY OATA
C
      COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP
C
C *** COMMON TO CONTAIN HARMONICS COEFFICIENTS
C
      COMMON / HARCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH
C
C *** ISTOP IS A FLAG SET INTERACTIVELY IN ROTERM TO INOICATE ENO OF PRO
C
      ISTOP=0
C
C *** ROFILE WILL REAO OISK OATA FILE CREATEO BY PUF-2.  NOTE THAT NSR I
C     PUF-2 INPUT MUST BE OIVISIBLE BY THE NUMBER OF BLAOES, NBLAOE.
C
      CALL ROFILE
C
C *** INTERACTIVELY REAO PARAMETERS FOR TERMINAL SESSION
C
 905  FORMAT(/' ENTER NBLAOE, NTORL, IPLOT.'/
     &  ' FOR HELP, ENTER ZERO FOR NBLAOE.')
  30  WRITE(6,905)
      REAO(5,*) NBLAOE,NTORL,IPLOT
      IF (NBLAOE.LT.O) GOTO 20
      IF (NBLAOE.EQ.O) GOTO 90
      IF (IPLOT.EQ.2) CALL PIEOAT(ISTOP)
      IF (ISTOP.EQ.1) GOTO 20
C
```

```
C *** ROTERM GETS DATA INTERACTIVELY FROM TERMINAL AND COMPUTES
C     NECESSARY FIELD POINT GEOMETRY.
C
  10  CALL ROTERM(ISTOP)
C
C *** IF OPERATOR INDICATES END OF PROGRAM, ISTOP WILL BE SET TO 1
C
      IF (ISTOP.GT.0) GOTO 20
C
C *** RUN THROUGH FPSTEP FOR EACH "TIME STEP."
C
      DO 100 K=1,NSR
C
C *** FPSTEP CALCULATED VELOCITY INDUCED AT FIELD POINT BY ONE BLADE
C     AT "TIME STEP" K
C
 100  CALL FPSTEP(K)
C
C *** SUMVEL WILL SUM FIELD POINT VELOCITIES INDUCED BY EACH BLADE.
C
      CALL SUMVEL
C
C *** FOUCAL WILL GENERATE HARMONIC COEFFICIENTS FOR INDUCED FIELD POINT
C     VELOCITY AS PROPELLER ROTATES.
C
      CALL FOUCAL
C
C *** PLOTVL PLOTS INDUCED FIELD POINT VELOCITIES AS A FUNCTION OF BLADE
C     ROTATION.
C
 901  FORMAT(' AXIAL COORDINATE',4X,F5.3/' RADIAL COORDINATE',2X,F6.3/
     &  ' ANGULAR COORDINATE',2X,F5.1///' AXIAL HARMONIC',6X,F8.5/
     &  ' RADIAL HARMONIC',5X,F8.5/' TANGENTIAL HARMONIC ',F8.5)
      WRITE(6,901) XFP,RFP,TZFP,(AMP(1,J),J=1,3)
      IF (IPLOT.EQ.1) CALL PLOTVL(NBLADE,NSR,NH,TFP,TZFP,U,A,B,AMP,PH,
     &  XFP,RFP,IDENT)
      GOTO 10
 906  FORMAT(/' NBLADE < 0 => STOP'
     &        /8X,     '= 0 => HELP'/8X,'> 0 => NUMBER OF BLADES'/
     &  ' NTORL  = 0 => TOTAL EFFECT'/8X,'= 1 => THICKNESS ONLY'/
     &  8X,'= 2 => LOADING ONLY'/' IPLOT  = 0 => DATA TO TERMINAL'/
     &  8X,'= 1 => PLOT -- NOT IMPLEMENTED 05/02/82 -- RLJ'
     &  /8X,'= 2 => CREATE PIEWAKE DATA FILE'/)
  90  WRITE(6,906)
      GOTO 30
  20  STOP
      END
```

```
      SUBROUTINE FPVEL(K,XF,YF,ZF,VIX,VIY,VIZ)                          UFPO1010
C                                                                       UFPO1020
C *** COMMON TO CONTAIN GEOMETRIC PARAMETERS                            UFPO1030
C                                                                       UFPO1040
      COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),   UFPO1050
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),       UFPO1060
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH                     UFPO1070
C                                                                       UFPO1080
C *** COMMON TO CONTAIN SINGULARITY STRENGTHS                           UFPO1090
C                                                                       UFPO1100
      COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV   UFPO1110
     & ,GAM(9,60),GMEAN(9),OKFW(20),HBWX,HBWY,HBWZ                      UFPO1120
C                                                                       UFPO1130
C *** COMMON TO CONTAIN FIELO POINT OATA, NBLAOE, IOENT, NSR, ANO VSR   UFPO1140
C                                                                       UFPO1150
      COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,               UFPO1160
     &   OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPLOT,OIAM                   UFPO1170
C                                                                       UFPO1180
      DIMENSION UTX(20,2),UTY(20,2),UTZ(20,2),UBX(20),UBY(20),UBZ(20)   UFPO1190
     & ,VOFWX(20,9),VOFWY(20,9),VOFWZ(20,9)                             UFPO1200
C                                                                       UFPO1210
C *** INITIALIZE VARIABLES                                              UFPO1220
C                                                                       UFPO1230
      MIO=MN/2                                                          UFPO1240
      NVFW=NW                                                           UFPO1250
      UMPFW=1                                                           UFPO1260
      WAKX=O.                                                           UFPO1270
      WAKY=O.                                                           UFPO1280
      WAKZ=O.                                                           UFPO1290
      VIX=0.0                                                           UFPO1300
      VIY=0.0                                                           UFPO1310
      VIZ=0.0                                                           UFPO1320
      DO 1 M=1,MM                                                       UFPO1330
      OO 1 N=1,NN                                                       UFPO1340
C-----CHOROWISE VORTICES ON BLAOE---------------------------------------UFPO1350
      IF(NTORL.EQ.1) GO TO 8                                            UFPO1360
      CALL VORSEG(XF,YF,ZF,X(N,M),Y(N,M),Z(N,M),X(N+1,M),Y(N+1,M),      UFPO1370
     1  Z(N+1,M),CVX,CVY,CVZ,SX,SY,SZ,0)                                UFPO1380
C-----SPANWISE VORTICES ANO SOURCES ON BLAOE----------------------------UFPO1390
 8    CALL VORSEG(XF,YF,ZF,X(N,M),Y(N,M),Z(N,M),X(N,M+1),Y(N,M+1),      UFPO1400
     1  Z(N,M+1),SVX,SVY,SVZ,SX,SY,SZ,1)                                UFPO1410
      SBNM=SB(N,M)                                                      UFPO1420
      GBNM=GB(N,M,K)                                                    UFPO1430
      GTNM=GT(N,M,K)                                                    UFPO1440
      IF(NTORL.EQ.0) GO TO 9                                            UFPO1450
      IF(NTORL.EQ.1) GBNM=0.0                                           UFPO1460
      IF(NTORL.EQ.1) GTNM=0.0                                           UFPO1470
      IF(NTORL.EQ.2) SBNM=0.0                                           UFPO1480
 9    VIX=VIX+GBNM*SVX+SBNM*SX+GTNM*CVX                                 UFPO1490
      VIY=VIY+GBNM*SVY+SBNM*SY+GTNM*CVY                                 UFPO1500
      VIZ=VIZ+GBNM*SVZ+SBNM*SZ+GTNM*CVZ                                 UFPO1510
 1    CONTINUE                                                          UFPO1520
C-----SEPARATEO TIP VORTEX ON BLAOE-----------------------------         UFPO1530
```

```
      IF(NTORL.EO.1) GO TO 10                                           UFPO1540
      OO 2 N=1,NN                                                       UFPO1550
      CVX=O.O                                                           UFPO1560
      CVY=O.O                                                           UFPO1570
      CVZ=O.O                                                           UFPO1580
      NTEM=NN-N+1                                                       UFPO1590
      OO 3 LL=1,NTEM                                                    UFPO1600
      L1=LL+1                                                           UFPO1610
      CALL VORSEG(XF,YF,ZF,XTIP(N,LL),YTIP(N,LL),ZTIP(N,LL)             UFPO1620
     *  ,XTIP(N,L1),YTIP(N,L1),ZTIP(N,L1),VX,VY,VZ,SX,SY,SZ,O)          UFPO1630
      CVX=CVX+VX                                                        UFPO1640
      CVY=CVY+VY                                                        UFPO1650
 3    CVZ=CVZ+VZ                                                        UFPO1660
      VIX=VIX+GB(N,MM,K)*CVX                                            UFPO1670
      VIY=VIY+GB(N,MM,K)*CVY                                            UFPO1680
 2    VIZ=VIZ+GB(N,MM,K)*CVZ                                            UFPO1690
C-----TRANSITION WAKE----------------------------------------------------UFPO1700
      WAKX=O.                                                           UFPO1710
      WAKY=O.                                                           UFPO1720
      WAKZ=O.                                                           UFPO1730
      OO 35 N=1,100                                                     UFPO1740
      CALL VORSEG(XF,YF,ZF,XH(N,1),YH(N,1),ZH(N,1),XH(N+1,1),YH(N+1,1), UFPO1750
     1  ZH(N+1,1),VX,VY,VZ,SX,SY,SZ,O)                                  UFPO1760
      WAKX=WAKX+VX                                                      UFPO1770
      WAKY=WAKY+VY                                                      UFPO1780
   35 WAKZ=WAKZ+VZ                                                      UFPO1790
      CALL VORSEG(XF,YF,ZF,XH(1,1),O.,O.,XH(101,1),O.,O.,               UFPO1800
     &  HBWX,HBWY,HBWZ,SX,SY,SZ,O)                                      UFPO1810
      HBWX=-HBWX                                                        UFPO1820
      HBWY=-HBWY                                                        UFPO1830
      HBWZ=-HBWZ                                                        UFPO1840
  11  OO 71 M=1,MM                                                      UFPO1850
      IF(M.EQ.1) GO TO 72                                               UFPO1860
      OO 73 N=1,NW                                                      UFPO1870
      UTX(N,1)=UTX(N,2)                                                 UFPO1880
      UTY(N,1)=UTY(N,2)                                                 UFPO1890
   73 UTZ(N,1)=UTZ(N,2)                                                 UFPO1900
   72 OO 74 N=1,NW                                                      UFPO1910
      CALL VORSEG(XF,YF,ZF,XW(N,M),YW(N,M),ZW(N,M),XW(N,M+1),YW(N,M+1), UFPO1920
     1  ZW(N,M+1),UBX(N),UBY(N),UBZ(N),SX,SY,SZ,O)                      UFPO1930
      IF(M.GT.1) GO TO 75                                               UFPO1940
      CALL VORSEG(XF,YF,ZF,XW(N,M),YW(N,M),ZW(N,M),XW(N+1,M),YW(N+1,M), UFPO1950
     1  ZW(N+1,M),UTX(N,1),UTY(N,1),UTZ(N,1),SX,SY,SZ,O)                UFPO1960
   75 CALL VORSEG(XF,YF,ZF,XW(N,M+1),YW(N,M+1),ZW(N,M+1),XW(N+1,M+1),   UFPO1970
     1  YW(N+1,M+1),ZW(N+1,M+1),UTX(N,2),UTY(N,2),UTZ(N,2),SX,SY,SZ,O)  UFPO1980
   74 CONTINUE                                                          UFPO1990
      DO 76 N=1,NW                                                      UFPO2000
C                                                                       UFPO2010
C *** COMPUTE INOUCED VELOCITY FROM ONE FORCESHOE AT THIS RAOIUS        UFPO2020
C                                                                       UFPO2030
      BUG=UTX(N,2)-UTX(N,1)+UBX(N)                                      UFPO2040
      CAT=UTY(N,2)-UTY(N,1)+UBY(N)                                      UFPO2050
      OOG=UTZ(N,2)-UTZ(N,1)+UBZ(N)                                      UFPO2060
```

```
      IF(N.NE.NW) GO TO 77
      IF(M.NE.MID) GO TO 78
C
C *** ALL TRANSITION WAKE VORTICES DISAPPEAR EXCEPT MID, WHICH TURNS
C     INTO ULTIMATE WAKE VORTICES (TIP AND HUB) BY ASSUMING MEAN LOADING
C
      BUG=BUG+WAKX+HBWX
      CAT=CAT+WAKY+HBWY
      DOG=DOG+WAKZ+HBWZ
      GO TO 78
   77 BUG=BUG-UBX(N+1)
      CAT=CAT-UBY(N+1)
      DOG=DOG-UBZ(N+1)
   78 VOFWX(N,M)=BUG
      VOFWY(N,M)=CAT
      VOFWZ(N,M)=DOG
   76 CONTINUE
   92 DO 93 LN=1,NVFW
      KKM=K-LN
      IF (KKM.LE.0) KKM=KKM+NSR
      VIX=VIX+VOFWX(LN,M)*(GMEAN(M)
     1                  +DKFW(LN)*(GAM(M,KKM)-GMEAN(M)))
      VIY=VIY+VOFWY(LN,M)*(GMEAN(M)
     1                  +DKFW(LN)*(GAM(M,KKM)-GMEAN(M)))
 93   VIZ=VIZ+VOFWZ(LN,M)*(GMEAN(M)
     1                  +DKFW(LN)*(GAM(M,KKM)-GMEAN(M)))
   71 CONTINUE
  10  RETURN
      END
```

```
      SUBROUTINE RDFILE                                                 UFPO2360
C     ******************************************************            UFPO2370
C     *                                                    *            UFPO2380
C     *   SUBRDUTINE RDFILE READS DISK FILE14 DATA CREATED *            UFPO2390
C     *   BY PUF-2 AND XAVES DATA NEEDED BY FPUV IN        *            UFPO2400
C     *   CDMMONS GEDCOM AND SINCDM.  NOTE THAT NSR IN     *            UFPO2410
C     *   PUF-2 MUST BE DIVISIBLE BY THE NUMBER DF BLADES, *            UFPO2420
C     *   NBLADE.                                          *            UFPO2430
C     *                                                    *            UFPO2440
C     *                                                    *            UFPO2450
C     *   NN         =>                                    *            UFPO2460
C     *   MM         =>                                    *            UFPO2470
C     *   NW         => NUMBER OF VORTICES IN KEY WAKE     *            UFPO2480
C     *   NBLADE     => NUMBER DF PRDPELLER BLADES         *            UFPO2490
C     *   NVC        => NO DF CHDRD VDRTICES, DTHER BLAOES *            UFPO2500
C     *   NVS        => NO OF SPAN VDRTICES, DTHER BLADES  *            UFPO2510
C     *   NVW        => NO OF WAKE VORTICES, OTHER WAKE    *            UFPO2520
C     *   NSR        => NUMBER DF TIME STEPS/REVOLUTIDN    *            UFPO2530
C     *   NWK        => NUMBER DF RADII IN WAKE HARMONICS  *            UFPO2540
C     *                                                    *            UFPO2550
C     ******************************************************            UFPO2560
C                                                                       UFPO2570
C                                                                       UFPO2580
C *** COMMON TD CDNTAIN GEOMETRIC PARAMETERS                            UFPO2590
C                                                                       UFPO2600
      CDMMON / GEOCDM / MM,MN,NN,NW,NTDRL,X(12,10),Y(12,10),Z(12,10),   UFPO2610
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),       UFPO2620
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH                     UFPO2630
C                                                                       UFPO2640
C *** CDMMDN TD CONTAIN SINGULARITY STRENGTHS                           UFPO2650
C                                                                       UFPO2660
      COMMDN / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV   UFPO2670
     &  ,GAM(9,60),GMEAN(9),DKFW(20),HBWX,HBWY,HBWZ                     UFPO2680
C                                                                       UFPO2690
C *** CDMMDN TO CONTAIN FIELD POINT DATA, NBLADE, IDENT, NSR, AND VSR   UFPO2700
C                                                                       UFPO2710
      CDMMON / FPCOM / NBLADE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,               UFPO2720
     &   DBLADE,DTFP,IDENT(18),TFP(60),VSR,IPLOT,DIAM                   UFPO2730
C                                                                       UFPO2740
C                                                                       UFPO2750
      DIMENSION RZ(10),R(9),CHDCP(9),CHORZ(10),PHIB(90)                 UFPO2760
      DIMENSION XD(6,4,6),YO(6,4,6),ZD(6,4,6)                           UFPO2770
      DIMENSION RWK(11),AWA(16,11),BWA(16,11),AWR(16,11),BWR(16,11),    UFPO2780
     &  AWT(16,11),BWT(16,11)                                           UFPO2790
      DIMENSION XWO(5,4,6),YWD(5,4,6),ZWO(5,4,6)                        UFPO2800
      DIMENSION SBO(5,3),GBC(5,3,60),GAMWC(4,3,60),GTC(5,4,60),         UFPO2810
     &  GAMC(3,60),GMEANO(3),DKOW(4)                                    UFPO2820
      READ(14) AJ                                                       UFPO2830
      IF (AJ.GE.2.0) STDP                                               UFPO2840
      READ(14) ISTDY,NBLADE,MM,NN,NW,NVC,NVS,NVW,NSR,NWK                UFPO2850
C                                                                       UFPO2860
C *** SET CONSTANT FDR GRID                                             UFPO2870
C                                                                       UFPO2880
```

```
      NM=NN-1                                                           UFPO2890
      MN=MM+1                                                           UFPO2900
      NP=NN+1                                                           UFPO2910
      NZ=NW+1                                                           UFPO2920
      NVCP=NVC+1                                                        UFPO2930
      NVSP=NVS+1                                                        UFPO2940
      NBLO=NBLADE-1                                                     UFPO2950
C                                                                       UFPO2960
C *** SET CONSTANTS FOR UNSTEADY PROBLEM                                 UFPO2970
C                                                                       UFPO2980
      IF (ISTDY.EQ.1) NSR=1                                             UFPO2990
      NVWP=NVW+1                                                        UFPO3000
      NTB=NN*MM                                                         UFPO3010
C                                                                       UFPO3020
C *** READ KEY BLADE GEOMETRY                                            UFPO3030
C                                                                       UFPO3040
      READ(14) ((X(N,M),N=1,NP),M=1,MN),                                UFPO3050
     2         ((Y(N,M),N=1,NP),M=1,MN),                                UFPO3060
     3         ((Z(N,M),N=1,NP),M=1,MN),                                UFPO3070
     4         (RZ(M),M=1,MN),                                          UFPO3080
     5         (R(M),M=1,MM),                                           UFPO3090
     6         (CHOCP(M),M=1,MM),                                       UFPO3100
     7         (CHORZ(M),M=1,MN),                                       UFPO3110
     8         (PHIB(K),K=1,NTB)                                        UFPO3120
C                                                                       UFPO3130
C *** READ BLADE GEOMETRY OF OTHER BLADES                                UFPO3140
C                                                                       UFPO3150
      IF (NBLADE.LE.1) GOTO 11                                          UFPO3160
      READ(14) (((XO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),              UFPO3170
     2         (((YO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),              UFPO3180
     3         (((ZO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO)               UFPO3190
   11 CONTINUE                                                          UFPO3200
C                                                                       UFPO3210
C *** READ WAKE RADII AND WAKE HARMONICS                                 UFPO3220
C                                                                       UFPO3230
      READ(14) (RWK(M),M=1,NWK)                                         UFPO3240
      READ(14) ((AWA(I,M),I=1,16),M=1,NWK),                             UFPO3250
     2         ((BWA(I,M),I=1,16),M=1,NWK),                             UFPO3260
     3         ((AWR(I,M),I=1,16),M=1,NWK),                             UFPO3270
     4         ((BWR(I,M),I=1,16),M=1,NWK),                             UFPO3280
     5         ((AWT(I,M),I=1,16),M=1,NWK),                             UFPO3290
     6         ((BWT(I,M),I=1,16),M=1,NWK)                              UFPO3300
C                                                                       UFPO3310
C *** READ VARIOUS PROPELLER/WAKE CONSTANTS                              UFPO3320
C                                                                       UFPO3330
      READ(14) WAKE,RPM,DIAM,RH,CDRAG,VSR,UR,(IDENT(N),N=1,18)          UFPO3340
C                                                                       UFPO3350
C *** READ KEY WAKE GEOMETRY                                             UFPO3360
C                                                                       UFPO3370
      READ(14) ((XW(N,M),N=1,NZ),M=1,MN),                               UFPO3380
     2         ((YW(N,M),N=1,NZ),M=1,MN),                               UFPO3390
     3         ((ZW(N,M),N=1,NZ),M=1,MN),                               UFPO3400
     4         ((XH(N,K),N=1,101),K=1,NBLADE),                          UFPO3410
```

```
     5          ((YH(N,K),N=1, 101),K=1,NBLADE),                         UFP03420
     6          ((ZH(N,K),N=1, 101),K=1,NBLADE),                         UFP03430
     7          ((XTIP(N,L),L=1,NP),N=1,NN),                             UFP03440
     8          ((YTIP(N,L),L=1,NP),N=1,NN),                             UFP03450
     9          ((ZTIP(N,L),L=1,NP),N=1,NN),                             UFP03460
C                                                                        UFP03470
C *** READ KEY BLADE SINGULARITY STRENGTHS                               UFP03480
C                                                                        UFP03490
     *          ((SB(N,M),N=1,NN),M=1,MM),                               UFP03500
     1          (((GB(N,M,KK),N=1,NN),M=1,MM),KK=1,NSR),                 UFP03510
     2          (((GT(N,M,KK),N=1,NN),M=1,MN),KK=1,NSR),                 UFP03520
     3          ((GAM(M,KK),M=1,MM),KK=1,NSR)                            UFP03530
C                                                                        UFP03540
C *** OTHER WAKE GEOMETRY                                                UFP03550
C                                                                        UFP03560
      IF (NBLADE.LE.1) GOTO 12                                           UFP03570
      READ(14) (((XWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),              UFP03580
     2          (((YWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),             UFP03590
     3          (((ZWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),             UFP03600
C                                                                        UFP03610
C *** OTHER BLADE SINGULARITIES                                          UFP03620
C                                                                        UFP03630
     4          ((SBO(N,M),N=1,NVC),M=1,NVS),                            UFP03640
     5          (((GBC(N,M,KK),N=1,NVC),M=1,NVS),KK=1,NSR),              UFP03650
     6          (((GAMWC(N,M,KK),N=1,NVW),M=1,NVS),KK=1,NSR),            UFP03660
     7          (((GTC(N,M,KK),N=1,NVC),M=1,NVSP),KK=1,NSR),             UFP03670
     8          ((GAMC(M,KK),M=1,NVS),KK=1,NSR),                         UFP03680
     9          (GMEAN(M),M=1,MM),                                       UFP03690
     *          (GMEANO(M),M=1,NVS),                                     UFP03700
     1          (DKFW(N),N=1,NW),                                        UFP03710
     2          (DKOW(N),N=1,NVW)                                        UFP03720
  12  CONTINUE                                                           UFP03730
  99  GOTO 100                                                           UFP03740
C                                                                        UFP03750
C &&& DIAGNOSTICS &&&&&&&&&&&&&&&&&&&&&&&                                UFP03760
C                                                                        UFP03770
 901  FORMAT(16I5)                                                       UFP03780
 902  FORMAT(8F10.5)                                                     UFP03790
 903  FORMAT(18A4)                                                       UFP03800
      WRITE(1,902) AJ                                                    UFP03810
      WRITE(1,901) ISTDY,NBLADE,MM,NN,NW,NVC,NVS,NVW,NSR,NWK             UFP03820
      WRITE(1,902) ((X(N,M),N=1,NP),M=1,MN),                             UFP03830
     2          ((Y(N,M),N=1,NP),M=1,MN),                                UFP03840
     3          ((Z(N,M),N=1,NP),M=1,MN),                                UFP03850
     4          (RZ(M),M=1,MN),                                          UFP03860
     5          (R(M),M=1,MM),                                           UFP03870
     6          (CHOCP(M),M=1,MM),                                       UFP03880
     7          (CHORZ(M),M=1,MN),                                       UFP03890
     8          (PHIB(K),K=1,NTB)                                        UFP03900
      IF (NBLADE.LE.1) GOTO 21                                           UFP03910
      WRITE(1,902) (((XO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),           UFP03920
     2          (((YO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO),              UFP03930
     3          (((ZO(N,M,K),N=1,NVCP),M=1,NVSP),K=1,NBLO)               UFP03940
```

```
 21   CONTINUE                                                          UFP03950
      WRITE(1,902) (RWK(M),M=1,NWK)                                     UFP03960
      WRITE(1,902) ((AWA(I,M),I=1,16),M=1,NWK),                         UFP03970
     3         ((BWA(I,M),I=1,16),M=1,NWK),                             UFP03980
     4         ((AWR(I,M),I=1,16),M=1,NWK),                             UFP03990
     5         ((BWR(I,M),I=1,16),M=1,NWK),                             UFP04000
     6         ((AWT(I,M),I=1,16),M=1,NWK),                             UFP04010
     7         ((BWT(I,M),I=1,16),M=1,NWK)                              UFP04020
      WRITE(1,902) WAKE,RPM,DIAM,RH,CDRAG,VSR,UR                        UFP04030
      WRITE(1,903) (IDENT(N),N=1,18)                                    UFP04040
      WRITE(1,902) ((XW(N,M),N=1,NZ),M=1,MN),                           UFP04050
     2         ((YW(N,M),N=1,NZ),M=1,MN),                               UFP04060
     3         ((ZW(N,M),N=1,NZ),M=1,MN),                               UFP04070
     4         ((XH(N,K),N=1,101),K=1,NBLADE),                          UFP04080
     5         ((YH(N,K),N=1,101),K=1,NBLADE),                          UFP04090
     6         ((ZH(N,K),N=1,101),K=1,NBLADE),                          UFP04100
     7         ((XTIP(N,L),L=1,NP),N=1,NN),                             UFP04110
     8         ((YTIP(N,L),L=1,NP),N=1,NN),                             UFP04120
     9         ((ZTIP(N,L),L=1,NP),N=1,NN)                              UFP04130
      WRITE(1,902) ((SB(N,M),N=1,NN),M=1,MM),                           UFP04140
     2         (((GB(N,M,KK),N=1,NN),M=1,MM),KK=1,NSR),                 UFP04150
     3         (((GT(N,M,KK),N=1,NN),M=1,MN),KK=1,NSR),                 UFP04160
     4         ((GAM(M,KK),M=1,MM),KK=1,NSR)                            UFP04170
      IF (NBLADE.LE.1) GOTO 22                                          UFP04180
      WRITE(1,902) (((XWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO),         UFP04190
     2         (((YWD(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLD),             UFP04200
     3         (((ZWO(N,M,K),N=1,NVWP),M=1,NVSP),K=1,NBLO)              UFP04210
      WRITE(1,902) ((SBD(N,M),N=1,NVC),M=1,NVS),                        UFP04220
     2         (((GBC(N,M,KK),N=1,NVC),M=1,NVS),KK=1,NSR),              UFP04230
     3         (((GAMWC(N,M,KK),N=1,NVW),M=1,NVS),KK=1,NSR),            UFP04240
     4         (((GTC(N,M,KK),N=1,NVC),M=1,NVSP),KK=1,NSR),             UFP04250
     5         ((GAMC(M,KK),M=1,NVS),KK=1,NSR),                         UFP04260
     6         (GMEAN(M),M=1,MM),                                       UFP04270
     7         (GMEANO(M),M=1,NVS),                                     UFP04280
     8         (DKFW(N),N=1,NW),                                        UFP04290
     9         (DKOW(N),N=1,NVW)                                        UFP04300
 22   CONTINUE                                                          UFP04310
100   RETURN                                                            UFP04320
      END                                                               UFP04330
```

```
      SUBROUTINE ROTERM(ISTOP)
C     *****************************************************
C     *                                                   *
C     *   SUBROUTINE ROTERM READS IN BLADE AND GEOMETRY   *
C     *   DATA INTERACTIVELY AND COMPUTES NECESSARY FIELD  *
C     *   POINT GEOMETRY.                                 *
C     *                                                   *
C     *   NBLADE      => NUMBER OF BLADES                 *
C     *   NTORL   = 0 => CONSIDER THICKNESS AND LOADING   *
C     *           = 1 => CONSIDER THICKNESS ONLY          *
C     *           = 2 => CONSIDER LOADING ONLY            *
C     *   XFP         => AXIAL FIELD POINT COORDINATE,    *
C     *                  POSITIVE DOWNSTREAM              *
C     *   RFP         => RADIAL FIELD POINT COORDINATE    *
C     *   TZFP        => INITAL FIELD POINT ANGLE IN DEG.  *
C     *   YBAR,ZBAR   => Y AND Z FIELD POINT COORDINATES  *
C     *   TH          => ANGLE FROM TIME STEP INPUT       *
C     *   YB,ZB       => Y,Z FIELD PT COORD, BEFORE TIME ST *
C     *                                                   *
C     *****************************************************
C
C *** COMMON TO CONTAIN GEOMETRIC PARAMETERS
C
      COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH
C
C *** COMMON TO CONTAIN FIELD POINT DATA, NBLADE, IDENT, NSR, AND VSR
C
      COMMON / FPCOM / NBLADE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,
     &   DBLADE,DTFP,IDENT(18),TFP(60),VSR,IPLOT,DIAM
C
C
 901  FORMAT(/' ENTER XFP, RFP, TZFP.'/
     &   ' FOR HELP, ENTER -11. FOR XFP.')
  10  WRITE(6,901)
      READ(5,*) XFP,RFP,TZFP
      IF (XFP.LT.-10.) GOTO 90
      IF (TZFP.LT.0.0) GOTO 70
      IF (RFP.GT.0.) GOTO 60
  20  WRITE(6,902)
 902  FORMAT(/' ENTER NBW,N,M,KK'/' FOR HELP, ENTER ZERO FOR NBW.')
      READ(5,*) NBW,N,M,KK
      TH=6.283185*FLOAT(KK-1)/FLOAT(NSR)
      IF (NBW.EQ.0) GOTO 80
C
C *** FIELD POINT CENTERED ON BLADE GRID.
C
      IF(NBW.EQ.1) CALL FPGRID(X,Y,Z,N,M,KK,12)
C
C *** FIELD POINT CENTERED ON TRANSITION WAKE GRID.
C
      IF(NBW.EQ.2) CALL FPGRID(XW,YW,ZW,N,M,KK,21)
```

```
      IF(NBW.NE.1.ANO.NBW.NE.2) GOTO 10
 60   OTFP=360.0/FLOAT(NSR)
      RETURN
 70   ISTOP=1
      RETURN
 80   WRITE(6,903)
903   FORMAT(/' NBW = 0 => HELP'/5X,'= 1 => BLAOE WAKE GRID'/5X,
     &  '= 2 => TRANSITION WAKE GRIO'/' N',7X,'=> CHOROWISE COOROINATE'/
     &  ' M',7X,'=> SPANWISE COORDINATE'/' KK',6X,
     &  '=> TIME STEPS ROTATEO + 1'/)
      GOTO 20
 90   WRITE(6,904)
904   FORMAT(/' XFP    <-10=> HELP'/8X,
     &  '>-10=> AXIAL FIELO POINT COOROINATE, POSITIVE OOWNSTREAM'/
     &  ' RFP    < 0 => SELECT COORDINATE USING BLAOE OR WAKE GRIO'/
     &  8X,'> 0 => RAOIAL FIELD POINT COORDINATE'/
     &  ' TZFP   < 0 => STOP'/8X,'> 0 => ANGULAR FILE POINT COOROINATE'
     &  /)
      GOTO 10
      END
```

```
      SUBROUTINE FPGRIO(XX.YY.ZZ.N.M.KK.NDIM)
C     ******************************************************
C     *                                                    *
C     *   SUBRDUTINE GRID CALCULATES X.R. AND THETA FIELD  *
C     *   PDINT GEDMETRY COMPONENTS GIVEN GRIO VECTORS     *
C     *   X.Y. ANO Z FOR BLAOE OR WAKE. GRIO CDDROINATES   *
C     *   N AND M. AND TIME STEP KK.                       *
C     *                                                    *
C     *   XX.YY.ZZ  => GRIO GEDMETRY VECTDRS               *
C     *   N,M       => GRIO INDICES SELECTED FOR VELDCITY  *
C     *   KK        => TIME STEP. NUMBER OF ANGLE STEPS + 1*
C     *   XFP.YBAR.ZBAR.TZFP.RFP                           *
C     *             => X.Y.Z.THETA.RADIAL FIELD PT CODRO   *
C     *   OBLADE    => ANGLE BETWEEN BLAOES                *
C     *   TH        => ANGLE OF TIME STEP                  *
C     *   NOIM      => FIRST OIMENSION DF GRIDS PASSED     *
C     *                                                    *
C     ******************************************************
C
C *** CDMMDN TO CONTAIN FIELD PDINT DATA. NBLAOE. IDENT. NSR. ANO VSR
C
      CDMMDN / FPCDM / NBLADE,NSR.XFP.RFP.TZFP.YBAR.ZBAR.
     &   OBLAOE.DTFP.IDENT(18).TFP(60).VSR.IPLOT.DIAM .
C
      OIMENSION XX(NDIM.10).YY(NDIM.10).ZZ(NOIM.10)
C
C *** COMPUTE X.Y.Z CDDROINATES IN ZERO ANGLE PDSITION
C
      TH=6.283185*FLDAT(KK-1)/FLDAT(NSR)
      THDEG=TH*360./6.283185
      OBLAOE=360.0/FLDAT(NBLAOE)
      XFP=0.25*(XX(N,M) + XX(N+1,M) + XX(N,M+1) + XX(N+1,M+1))
      YB =0.25*(YY(N,M) + YY(N+1,M) + YY(N,M+1) + YY(N+1,M+1))
      ZB =0.25*(ZZ(N,M) + ZZ(N+1,M) + ZZ(N,M+1) + ZZ(N+1,M+1))
C
C *** CONVERT TD CYLINDRICAL CDDRDINATES
C
      RFP=SQRT(YB*YB+ZB*ZB)
      TZFP=ATAN2(ZB.YB)/1.7453293E-02
      TZFP=TZFP+THDEG
C
C *** BRANCH IF GENERATING PIEWAKE OATA
C
      IF (IPLDT.EQ.2) GDTD 200
      TZFP=180.0/FLDAT(NBLAOE) - ATAN2(ZB.YB)/1.7453293E-02
C
C *** KEEP TZFP BETWEEN O AND DNE BLAOE ANGLE
C
      OD 100 K=1,NBLAOE
      IF(TZFP.LT.0.0) TZFP=TZFP+OBLAOE
      IF(TZFP.GT.DBLAOE) TZFP=TZFP-DBLAOE
  100 CONTINUE
C
```

```
C *** ROTATE BLADE FOR PIEWAKE DATA                                     UFPO5600
C                                                                       UFPO5610
 200  YBAR=YB*COS(TH) - ZB*SIN(TH)                                      UFPO5620
      ZBAR=YB*SIN(TH) + ZB*COS(TH)                                      UFPO5630
 300  RETURN                                                            UFPO5640
      END                                                               UFPO5650
```

```
      SUBROUTINE FPSTEP(K)                                              UFP05660
C     ***************************************************               UFP05670
C     *                                                 *               UFP05680
C     *   FPSTEP IS A SUBROUTINE TO CALCULATE THE FIELO *               UFP05690
C     *   POINT VELOCITIES INOUCEO BY ONE BLAOE AT "TIME *              UFP05700
C     *   STEP", K.  THE "OIRTY WORK" IS OONE IN        *               UFP05710
C     *   SUBROUTINE FPVEL.                             *               UFP05720
C     *                                                 *               UFP05730
C     *   TFP      => ANGULAR FIELD POINT COOROINATE, OEG. *            UFP05740
C     *   TRAO     => ANGULAR FIELO POINT COOROINATE, RAO. *            UFP05750
C     *   YFP,ZFP  => Y ANO Z FIELO POINT COOROINATES   *               UFP05760
C     *   VIX,VIY,VIZ,VIR,VIT => INOUCEO VELOCITY COMPONENT*            UFP05770
C     *                                                 *               UFP05780
C     ***************************************************               UFP05790
C                                                                       UFP05800
C *** COMMON TO CONTAIN GEOMETRIC PARAMETERS                            UFP05810
C                                                                       UFP05820
      COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),   UFP05830
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),       UFP05840
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH                     UFP05850
C                                                                       UFP05860
C *** COMMON TO CONTAIN SINGULARITY STRENGTHS                           UFP05870
C                                                                       UFP05880
      COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV   UFP05890
     &  ,GAM(9,60),GMEAN(9),OKFW(20),HBWX,HBWY,HBWZ                     UFP05900
C                                                                       UFP05910
C *** COMMON TO CONTAIN FIELD POINT OATA, NBLAOE, IOENT, NSR, ANO VSR   UFP05920
C                                                                       UFP05930
      COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,               UFP05940
     &   OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPLOT,OIAM                   UFP05950
C                                                                       UFP05960
C *** COMMON TO CONTAIN VELOCITY OATA                                   UFP05970
C                                                                       UFP05980
      COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP           UFP05990
C                                                                       UFP06000
C *** CALCULATE CARTESIAN FIELO POINT COOROINATES TO FEEO FPVEL         UFP06010
C                                                                       UFP06020
      TFP(K)=FLOAT(K-1)*OTFP-TZFP                                       UFP06030
      TRAD=TFP(K)*1.7453293E-02                                         UFP06040
      COSFP=COS(TRAD)                                                   UFP06050
      SINFP=SIN(TRAO)                                                   UFP06060
      YFP=RFP*COSFP                                                     UFP06070
      ZFP=RFP*SINFP                                                     UFP06080
C                                                                       UFP06090
C *** FPVEL WILL CALCULATE VIX, VIY, ANO VIZ FOR FIELO POINT            UFP06100
C                                                                       UFP06110
      CALL FPVEL(K,XFP,YFP,ZFP,VIX(K),VIY,VIZ)                          UFP06120
      VIR(K)=VIY*COSFP + VIZ*SINFP                                      UFP06130
      VIT(K)=VIZ*COSFP - VIY*SINFP                                      UFP06140
      RETURN                                                            UFP06150
      END                                                               UFP06160
```

```
      SUBROUTINE SUMVEL
C     *******************************************************
C     *                                                     *
C     *   SUBROUTINE SUMVEL SUMS VELOCITIES FROM EACH       *
C     *   BLADE, CALCULATED IN FPSTEP, TO PRODUCE THE       *
C     *   TOTAL PROPELLER-INDUCED FIELO POINT VELOCITY.     *
C     *                                                     *
C     *   VIX,VIR,VIT => VELOCITY COMPONENTS INOUCED BY     *
C     *                  ONE BLAOE AT "TIME STEP" K         *
C     *   U(N,J)      => VELOCITY INOUCED BY ALL BLAOES     *
C     *                  AT "TIME STEP" N                   *
C     *           J=0 => AXIAL COMPONENT                    *
C     *           J=1 => RAOIAL COMPONENT                   *
C     *           J=2 => TANGENTIAL COMPONENT               *
C     *   NSAMP       => NUMBER OF "TIME STEPS" IN ONE      *
C     *                  BLAOE ANGLE (= NUMBER OF           *
C     *                  U VELOCITIES COMPUTED.)            *
C     *                                                     *
C     *******************************************************
C
C *** COMMON TO CONTAIN FIELO POINT DATA, NBLAOE, IDENT, NSR, ANO VSR
C
      COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,
     &   OBLAOE,OTFP,IDENT(18),TFP(60),VSR,IPLOT,OIAM
C
C *** COMMON TO CONTAIN VELOCITY DATA
C
      COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP
C
C *** AT ANY GIVEN REAL TIME, EACH BLAOE CAN BE CONSIDERED TO HAVE
C     TRAVELLEO THROUGH A CERTAIN NUMBER OF "TIME STEPS".
C     BY SUMMING VIX, VIR, ANO VIT AT BLAOE RATE, WE CAN INCLUOE
C     THE VELOCITIES INDUCED BY EACH BLAOE INTO ONE VELOCITY
C     INOUCED BY THE ENTIRE PROPELLER.
C
      NSAMP=NSR/NBLAOE
      DO 20 N=1,NSAMP
      OO 10 J=1,3
   10 U(N,J)=0.0
      DO 20 K=1,NBLAOE
      L=N+(K-1)*NSAMP
      U(N,1)=U(N,1)+VIX(L)
      U(N,2)=U(N,2)+VIR(L)
   20 U(N,3)=U(N,3)+VIT(L)
C
C *** NOW CONVERT FROM REFERENCE VELOCITY TO SHIP VELOCITY
C
      OO 30 N=1,NSAMP
      OO 30 J=1,3
   30 U(N,J)=U(N,J)/VSR
      RETURN
      END
```

```
      SUBROUTINE FOUCAL
C     *****************************************************
C     *                                                   *
C     *   SUBROUTINE FOUCAL TAKES THE PROPELLER INOUCED   *
C     *   VELOCITIES CALCULATEO IN SUMVEL AND PERFORMS    *
C     *   A HARMONIC ANALYSIS ON THEM.  IT CALCULATES     *
C     *   FOURIER COEFFICIENTS FOR BOTH COSINE (A) ANO    *
C     *   SINE (B) ANO ALSO COSINE AMPLITUOE (AMP) AND    *
C     *   PHASE ANGLE (PH).                               *
C     *                                                   *
C     *   NH              => NUMBER OF HARMONICS CALCULATED *
C     *   A(N,J),B(N,J)  => FOURIER COEFFICIENTS          *
C     *                      A => COSINE COEFFICIENTS     *
C     *                      B => SINE COEFFICIENTS       *
C     *                      N => HARMONIC NUMBER UP TO 15 *
C     *                  J = 1 => AXIAL VELOCITY          *
C     *                    = 2 => RAOIAL VELOCITY         *
C     *                    = 3 => TANGENTIAL VELOCITY     *
C     *   AMP(N,J)       => FOURIER COEFFICIENT AMPLITUOE *
C     *                      N AND J SUBSCRIPTS AS FOR A, B *
C     *   PH(N,J)        => PHASE ANGLE ASSOCIATEO WITH AMP *
C     *   NFB            => NUMBER OF FIELO POINTS / BLAOE *
C     *                                                   *
C     *****************************************************
C
C *** COMMON TO CONTAIN FIELO POINT OATA, NBLAOE, IOENT, NSR, ANO VSR
C
      COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,
     &   OBLAOE,OTFP,IOENT(18),TFP(60),VSR,IPLOT,OIAM
C
C *** COMMON TO CONTAIN VELOCITY DATA
C
      COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP
C
C *** COMMON TO CONTAIN HARMONICS COEFFICIENTS
C
      COMMON / HARCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH
C
C *** FIRST CALCULATE A AND B
C
      NFB=NSR/NBLAOE
      NHMAX=NFB/2+1
      NH=MINO(NHMAX,15)
      STEP=6.283185/NFB
      ANFB=2.0/NFB
      BNFB=1.0/NFB
      00 11 K=1,NH
      DO 10 J=1,3
      A(K,J)=0.0
   10 B(K,J)=0.0
      DO 9 N=1,NFB
      T=((N-1)*(K-1))*STEP
      ST=SIN(T)
```

```
      CT=COS(T)                                                         UFP07220
      DO 9 J=1,3                                                        UFP07230
      A(K,J)=A(K,J)+U(N,J)*CT                                           UFP07240
 9    B(K,J)=B(K,J)+U(N,J)*ST                                           UFP07250
C                                                                       UFP07260
C *** CONVERT A AND B INTO AMP AND PH                                   UFP07270
C                                                                       UFP07280
      DO 11 J=1,3                                                       UFP07290
      IF(K.EQ.1) A(1,J)=A(1,J)*BNFB                                     UFP07300
      IF(K.EQ.NHMAX) GO TO 12                                           UFP07310
      IF(K.GT.1) A(K,J)=A(K,J)*ANFB                                     UFP07320
      IF(K.GT.1) B(K,J)=B(K,J)*ANFB                                     UFP07330
      GO TO 13                                                          UFP07340
 12   A(K,J)=A(K,J)*BNFB                                                UFP07350
      B(K,J)=0.0                                                        UFP07360
 13   AMP(K,J)=SQRT(A(K,J)**2+B(K,J)**2)                                UFP07370
      PH(K,J)=ATAN2(A(K,J),B(K,J))*57.296                               UFP07380
      IF(PH(K,J).LT.0.0) PH(K,J)=PH(K,J)+360.0                          UFP07390
      PH(K,J)=PH(K,J)/NBLADE                                            UFP07400
      PH(1,J)=0.0                                                       UFP07410
      AMP(1,J)=A(1,J)                                                   UFP07420
 11   CONTINUE                                                          UFP07430
      RETURN                                                            UFP07440
      END                                                               UFP07450
```

```
      SUBROUTINE PIEDAT(ISTOP)                                          UFP07460
C     ***********************************************************       UFP07470
C     *                                                         *       UFP07480
C     *   SUBROUTINE PIEOAT GENERATES OATA FROM INPUT TO        *       UFP07490
C     *   AXIAL VELOCITY DUTPUT TO BE USEO AS INPUT FOR         *       UFP07500
C     *   UNSTEAOY PIEWAKE.  IT USES BLADE GRID GEOMETRY        *       UFP07510
C     *   TD CALCULATE NSR FIELD POINT AXIAL TIME-AVERAGED *            UFP07520
C     *   VELDCITIES AT FIVE RADII ALDNG THE LEADING EDGE. *            UFP07530
C     *   IT THEN CALCULATES NSR FIELD POINT AXIAL TIME-        *       UFP07540
C     *   AVERAGED VELOCITIES AT 1.1 TIMES PROPELLER RAOIUS*            UFP07550
C     *                                                         *       UFP07560
C     *   N           => CHORDWISE GRID CDORDINATE              *       UFP07570
C     *   M           => SPANWISE GRID CDORDINATE               *       UFP07580
C     *   K,KK        => ANGLE INCREMENTS + 1                   *       UFP07590
C     *   ISTDP=1     => STDP FPUV UPDN RETURN TD MAIN PROG     *       UFP07600
C     *   NSR         => NUMBER TIME STEPS IN REVDLUTIDN        *       UFP07610
C     *   NR          => RADIUS NUMBER                          *       UFP07620
C     *   XFP,YBAR,ZBAR,RFP,TZFP                                *       UFP07630
C     *               => X,Y,Z,RAOIAL,THETA FIELO PT COORD      *       UFP07640
C     *   THETA(1-5)=> ANGLES 5 TIME STEPS BEFDRE ZERO          *       UFP07650
C     *                  BLAOE ANGLE                            *       UFP07660
C     *   THETA(6 - NSR+5) => ANGLES IN ONE BLAOE REV.          *       UFP07670
C     *   THETA(NSR+6 - NSR+20) => ANGLES 15 TIME STEPS         *       UFP07680
C     *                  AFTER ZERD BLAO ANGLE.                 *       UFP07690
C     *   VX(K)       => AXIAL VELDCITỸ AT ANGLE THETA(K)       *       UFP07700
C     *   UA(KK,M)    => SMOOTH AXIAL VELOCITY AT RADIUS M,     *       UFP07710
C     *                  ANGLE KK                               *       UFP07720
C     *                                                         *       UFP07730
C     ***********************************************************       UFP07740
C                                                                       UFP07750
C                                                                       UFP07760
C                                                                       UFP07770
C *** CDMMON TD CONTAIN GEOMETRIC PARAMETERS                            UFP07780
C                                                                       UFP07790
      COMMON / GEDCDM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),   UFP07800
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),       UFP07810
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH                     UFP07820
C                                                                       UFP07830
C *** COMMON TD CONTAIN SINGULARITY STRENGTHS                           UFP07840
C                                                                       UFP07850
      COMMON / SINCDM / GT(10,10,60),G8(10,9,60),GTW(10),S8(10,9),GTV   UFP07860
     &  ,GAM(9,60),GMEAN(9),DKFW(20),HBWX,HBWY,HBWZ                     UFP07870
C                                                                       UFP07880
C *** COMMDN TD CONTAIN FIELD POINT DATA, NBLADE, IDENT, NSR, ANO VSR   UFP07890
C                                                                       UFP07900
      COMMON / FPCDM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,               UFP07910
     &   DBLADE,DTFP,IDENT(18),TFP(60),VSR,IPLOT,DIAM                   UFP07920
C                                                                       UFP07930
C *** COMMON TD CONTAIN VELOCITY OATA                                   UFP07940
C                                                                       UFP07950
      COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP           UFP07960
C                                                                       UFP07970
C *** COMMON TD CDNTAIN HARMDNICS COEFFICIENTS                          UFP07980
```

```
C                                                                       UFPO7990
      COMMON / HARCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH           UFPO8000
C                                                                       UFPO8010
C                                                                       UFPO8020
      DIMENSION THETA(80),VX(80),UA(60,9),RWK(9)                        UFPO8030
      N=1                                                               UFPO8040
      NR=1                                                              UFPO8050
C                                                                       UFPO8060
C *** CALCULATE VELOCITIES AT FIVE BLAOE GRID RADII                     UFPO8070
C                                                                       UFPO8080
 102  00 300 M=3,9,2                                                    UFPO8090
      NR=NR+1                                                           UFPO8100
C                                                                       UFPO8110
C *** CALCULATE VELOCITIES AT NSR ANGLES FOR EACH RADIUS                UFPO8120
C                                                                       UFPO8130
      00 200 KK=1,NSR                                                   UFPO8140
C                                                                       UFPO8150
C *** FIELD POINT CENTERED ON BLADE GRIO                                UFPO8160
C                                                                       UFPO8170
      CALL FPGRIO(X,Y,Z,N,M,KK,12)                                      UFPO8180
      IF (M.EQ.3) X1=XFP                                                UFPO8190
      KK15=KK+15                                                        UFPO8200
      THETA(KK15)=TZFP                                                  UFPO8210
      RWK(NR)=RFP                                                       UFPO8220
      DTFP=360./FLOAT(NSR)                                              UFPO8230
C                                                                       UFPO8240
C *** CALCULATE AXIAL VELOCITY FOR FIELO POINT AT RAOIUS RWK(NR) ANO    UFPO8250
C *** ANGLE THETA(KK15).  STORE THIS VELOCITY IN VX.                    UFPO8260
C                                                                       UFPO8270
 200  CALL AMPCAL(VX(16),KK)                                            UFPO8280
      NSR15=NSR+15                                                      UFPO8290
C                                                                       UFPO8300
C *** STORE SMOOTHEO VELOCITIES IN NR-TH COLUMN OF UA                   UFPO8310
C                                                                       UFPO8320
 300  CALL SMOOTH(NSR,NR,VX,THETA,UA)                                   UFPO8330
C                                                                       UFPO8340
C *** CALCULATE NSR VELOCITIES AT PROPELLER HUB                         UFPO8350
C                                                                       UFPO8360
      NRHUB=1                                                           UFPO8370
      XFP=1.5*X1                                                        UFPO8380
      RWK(NRHUB)=RH                                                     UFPO8390
C                                                                       UFPO8400
C *** STEP THROUGH NSR ANGLES AT HUB                                    UFPO8410
C                                                                       UFPO8420
      DO 100 KK=1,NSR                                                   UFPO8430
      KK15=KK+15                                                        UFPO8440
      THETA(KK15)=360.*FLOAT(KK-1)/FLOAT(NSR)                           UFPO8450
      TZFP=THETA(KK15)                                                  UFPO8460
 100  CALL AMPCAL(VX(16),KK)                                            UFPO8470
      NSR15=NSR+15                                                      UFPO8480
C                                                                       UFPO8490
C *** NOW STORE SMOOTHED HUB VELOCITIES IN FIRST COLUMN OF UA           UFPO8500
C                                                                       UFPO8510
```

```
      CALL SMOOTH(NSR,NRHUB,VX,THETA,UA)
C
C *** NOW CALCULATE VELOCITIES AT A NEXT RADII OUTSIDE THE PROPELLER
C *** DISK.  USE XFP OF PROPELLER GRID CASE OF N=1, M=9.
C
C &&& DIAGNOSTICS &&&&&&&&&&&&&&&&
C
C     GOTO 101
C
C *** CALCULATE NSR VELOCITIES AT 1.1,1.4, AND 1.7 PROP RADII
C
      DO 500 KKK=1,7,3
      RFP=1.0+FLOAT(KKK)/10.
      NR=NR+1
      RWK(NR)=RFP
C
C *** CALCULATE NSR VELOCITIES AT THESE RADII
C
      DO 400 KK=1,NSR
      KK15=KK+15
      THETA(KK15)=360.*FLOAT(KK-1)/FLOAT(NSR)
      TZFP=THETA(KK15)
 400  CALL AMPCAL(VX(16),KK)
C
C *** NOW STORE SMOOTHED VELOCITIES IN NR-TH COLUMN OF UA
C
 500  CALL SMOOTH(NSR,NR,VX,THETA,UA)
C
C *** WRITE(OUTPUT IN FORMAT FOR UNSTEADY PIEWAKE
C
 901  FORMAT(8F10.5)
 902  FORMAT(16I5)
 101  WRITE(8,902) NR
      WRITE(8,901) (RWK(M),M=1,NR)
      WRITE(8,901) ((UA(KK,M),KK=1,NSR),M=1,NR)
C
C *** STOP FPUV UPON RETURN TO MAIN PROGRAM
C
      ISTOP=1
      RETURN
      END
```

```
      SUBROUTINE AMPCAL(VX,KK)
C     *****************************************************
C     *                                                   *
C     *   SUBROUTINE AMPCAL WILL TAKE FIELO POINT GEOMETRY *
C     *   CALCULATED IN PIEOAT ANO CALCULATE THE INDUCEO   *
C     *   TIME-AVERAGED AXIAL VELOCITY.  IT WILL STORE THAT*
C     *   VELOCITY IN VX(KK).                              *
C     *                                                   *
C     *   NSR        => NUMBER OF ANGLES PER REVOLUTION    *
C     *   K          => ANGLE INCREMENTS + 1               *
C     *   VX(KK)     => WHERE IME-AVERAGEO AXIAL VEL STORED *
C     *   AMP(1,1)   => ZEROETH-HARMONIC AXIOAL VELOCITY   *
C     *                 COMPUTED BY FOUCAL                 *
C     *                                                   *
C     *****************************************************
C
C
C
C *** COMMON TO CONTAIN GEOMETRIC PARAMETERS
C
      COMMON / GEOCOM / MM,MN,NN,NW,NTORL,X(12,10),Y(12,10),Z(12,10),
     &   XTIP(10,11),YTIP(10,11),ZTIP(10,11),XH(101,7),YH(101,7),
     &   ZH(101,7),XW(21,10),YW(21,10),ZW(21,10),RH
C
C *** COMMON TO CONTAIN SINGULARITY STRENGTHS
C
      COMMON / SINCOM / GT(10,10,60),GB(10,9,60),GTW(10),SB(10,9),GTV
     &  ,GAM(9,60),GMEAN(9),OKFW(20),HBWX,HBWY,HBWZ
C
C *** COMMON TO CONTAIN FIELO POINT DATA, NBLAOE, IDENT, NSR, AND VSR
C
      COMMON / FPCOM / NBLAOE,NSR,XFP,RFP,TZFP,YBAR,ZBAR,
     &   OBLAOE,OTFP,IDENT(18),TFP(60),VSR,IPLOT,OIAM
C
C *** COMMON TO CONTAIN VELOCITY OATA
C
      COMMON / VELCOM / VIX(60),VIT(60),VIR(60),U(60,3),NSAMP
C
C *** COMMON TO CONTAIN HARMONICS COEFFICIENTS
C
      COMMON / HARCOM / A(15,3),B(15,3),AMP(15,3),PH(15,3),NH
C
      OIMENSION VX(1)
C
C *** RUN THROUGH ANO FPSTEP FOR EACH TIME STEP
C
      OO 100 K=1,NSR
C
C *** FPSTEP CALCULATES VELOCITY INOUCED AT A FIELO POINT BY ONE
C *** BLAOE AT TIME STEP K
C
 100  CALL FPSTEP(K)
C
```

```
C *** SUMVEL SUMS FIELD POINT VELOCITIES INDUCED BY EACH BLADE
C
      CALL SUMVEL
C
C *** FOUCAL GENERATES HARMONIC COEFFICIENTS FOR INDUCED FIELD POINT
C *** VELOCITY AS PROPELLER ROTATES.
C
      CALL FOUCAL
C
C *** STORE TIME-AVERAGED (ZERO-HARMONIC) VELOCITY IN VX
C
      VX(KK)=AMP(1,1)
      RETURN
      END
```

```
      SUBROUTINE SMOOTH(NSR,NR,VX,THETA,UA)                             UFP09610
C     ****************************************************************  UFP09620
C     *                                                              *  UFP09630
C     *   SUBROUTINE SMOOTH REAOS INOUCED VELOCITIES AT ONE*            UFP09640
C     *   RAOIUS FROM PIEDAT.  IT CALCULATES BY                      *  UFP09650
C     *   INTERPOLATION A SERIES OF NSR EVENLY SPACEO                *  UFP09660
C     *   AXIAL INOUCED VELOCITIES SUITABLE FOR INPUT INTO           *  UFP09670
C     *   UNSTEADY PIEWAKE.                                          *  UFP09680
C     *                                                              *  UFP09690
C     *   NSR         => NUMBER OF ANGLES PER REVOLUTION             *  UFP09700
C     *   NR          => RAOIUS NUMBER 1=HUB, ETC.                   *  UFP09710
C     *   VX(1 - 5) => RAW VELOCITIES 5 STEPS BEFORE ZERO            *  UFP09720
C     *                BLADE ANGLE.                                  *  UFP09730
C     *   VX(6 - NSR+5) => RAW VELOCITIES IN ONE REVOLUTION*            UFP09740
C     *   VX(NSR+6 - NSR+20) => RAW VELOCITIES 15 TIME               *  UFP09750
C     *                TIME STEPS AFTER ONE REVOLUTION               *  UFP09760
C     *   THETA(K)    => ANGLE OF VELOCITY VX(K)                     *  UFP09770
C     *   UA(K,M)     => SMOOTH VEL AT RAOIUS M, ANGLE K             *  UFP09780
C     *   TH(K)       => EVENLY SPACED ANGLES OF UA(K,M)             *  UFP09790
C     *   NIN         => NUMBER OF ANGLES FED TO UGLYOK              *  UFP09800
C     *   KPOS        => K OF FIRST POS. ANGLE IN THETA(K)           *  UFP09810
C     *   KSTART,KSTOP => FIRST ANO LAST INOEX OF THETA(K)           *  UFP09820
C     *                AND VX(K) SENT TO UGLYOK                      *  UFP09830
C     *   AVX         => SPLINE CURVE COEFFICIENTS OF VX             *  UFP09840
C     *                                                              *  UFP09850
C     ****************************************************************  UFP09860
C                                                                       UFP09870
C                                                                       UFP09880
      DIMENSION VX(80),UA(60,9),THETA(80),TH(60),AVX(250)               UFP09890
C                                                                       UFP09900
C *** COPY LAST 15 STEPS IN REV INTO FIRST 15 ENTRIES IN VX ANO THETA   UFP09910
C                                                                       UFP09920
      00 100 K=1,15                                                     UFP09930
      KNSR=K+NSR                                                        UFP09940
      THETA(K)=THETA(KNSR)-360.                                         UFP09950
  100 VX(K)=VX(KNSR)                                                    UFP09960
C                                                                       UFP09970
C *** COPY FIRST 5 TIME STEPS IN REV INTO 5 ENTRIES AFTER REVOLUTION    UFP09980
C                                                                       UFP09990
      00 200 K=16,20                                                    UFP10000
      KNSR=K+NSR                                                        UFP10010
      THETA(KNSR)=THETA(K) + 360.                                       UFP10020
  200 VX(KNSR)=VX(K)                                                    UFP10030
C                                                                       UFP10040
C *** SET UP EVENLY SPACED ANGLES                                       UFP10050
C                                                                       UFP10060
      DO 300 K=1,NSR                                                    UFP10070
  300 TH(K)=360.*FLOAT(K-1)/FLOAT(NSR)                                  UFP10080
  901 FORMAT(8F10.5)                                                    UFP10090
  902 FORMAT(16I5)                                                      UFP10100
C                                                                       UFP10110
C *** FIND FIRST POSITIVE ANGLE, SAVE INDEX                             UFP10120
C                                                                       UFP10130
```

```
      DO 400 K=1,NSR                                                    UFP10140
      KPOS=K                                                            UFP10150
      IF (THETA(K).GT.0.) GOTO 500                                      UFP10160
  400 CONTINUE                                                          UFP10170
  500 KSTART=KPOS-1                                                     UFP10180
      NSR2=NSR/2                                                        UFP10190
      KSTOP=KPOS+NSR2+1                                                 UFP10200
      NIN=KSTOP-KSTART+1                                                UFP10210
C                                                                       UFP10220
C *** FINE SPLINE COEFFICIENTS AND SMOOTH VEL FOR FIRST 180 DEGREES     UFP10230
C                                                                       UFP10240
      CALL UGLYDK(NIN,1,1,THETA(KSTART),VX(KSTART),0.,0.,AVX)           UFP10250
      CALL EVALDK(NIN,NSR2,THETA(KSTART),TH(1),UA(1,NR),AVX)            UFP10260
C                                                                       UFP10270
C *** FIND SPLINE COEFFICIENTS AND SMOOTH VEL FOR LAST 180 DEGREES      UFP10280
C                                                                       UFP10290
      KSTART=KSTOP-2                                                    UFP10300
      KSTOP=KSTART+NIN-1                                                UFP10310
      NSR2P=NSR2+1                                                      UFP10320
      CALL UGLYDK(NIN,1,1,THETA(KSTART),VX(KSTART),0.,0.,AVX)           UFP10330
      CALL EVALDK(NIN,NSR2,THETA(KSTART),TH(NSR2P),UA(NSR2P,NR),AVX)    UFP10340
      RETURN                                                            UFP10350
      END                                                               UFP10360
```

** APPENDIX XI **

PIEWAKE Program Listing

```
C     THIS PROGRAM COMPUTES EFFECTIVE WAKES BY ONV METHOO
C
C     *******************************************************
C     *                                                     *
C     *   THE ORIGINAL PIEWAKE WAS WRITTEN BY R. VAN HOUTEN*
C     *   IN ABOUT 1981.  IT IS BASEO ON THE OET NORSKE     *
C     *   VERITAS METHOO OF COMPUTING EFFECTIVE WAKE.  THIS*
C     *   PROGRAM HAS BEEN MOOIFIEC BY R. JAMISON IN APRIL.*
C     *   1982, TO ACCOMOOATE UNSTEAOY PROPELLER-INDUCEO   *
C     *   AXIAL VELOCITIES FROM UFPV.  BASICALLY, BOTH     *
C     *   PROGRAMS OIVIOE THE WAKE INTO A NUMBER OF PIE-   *
C     *   SHAPEO SEGMENTS.  EACH SEGMENT IS ASSUMEO TO     *
C     *   CONTRACT IN AN AXISYMMETRIC MANNER AS GIVEN BY   *
C     *   T. HUANG.                                        *
C     *                                                     *
C     *   NP        => NUMBER OF PIE-SHAPEO SEGMENTS       *
C     *   NX        => NUMBER OF NOMINAL WAKE RAOII        *
C     *   RX(M)     => NOMINAL WAKE RAOIUS M (UP TO 11)    *
C     *   VO(K,M)   => NOMINAL AXIAL VELOCITY, SEG K, RAO M*
C     *   VT(K,M)   => NOM TANG VEL, SEG K, RAO M          *
C     *   VR(K,M)   => NOM RAO VEL, SEG K, RAO M           *
C     *   VTE,VRE   => VT & VR INTERPOLATEO AT EFF WAKE RAO*
C     *   UASEG(M)  => UA IN A PIE-SEGMENT.  ALSO USEO TO  *
C     *                STORE A PIE-SEGMENT OF VT           *
C     *   UX(M)     => VO IN A PIE-SEGMENT.  ALSO USEO TO  *
C     *                STORE A PIE-SEGMENT OF VR           *
C     *   AUX,AUASEG=> SPLINE COEFF. FOR VR,VT, RESP.      *
C     *   RE(M)     => RAOII OF EFFECTIVE WAKE VELOCITIES  *
C     *   THETA(K)  => ANGLE IN OEGREES OF PIE SEG K       *
C     *   NA        => NUMBER OF PROP-INO. AXIAL VEL RAOII *
C     *   RA(M)     => PROP-INO AXIAL VEL RAOIUS M         *
C     *   UA(K,M)   => PROP INO AXIAL VEL, SEG K, RAO M    *
C     *   VE(K,M)   => EFFECTIVE WAKE VEL, SEG K, RAD M    *
C     *   VL        => AVERAGE VELOCITY                    *
C     *   VLAVN     => VOLUMETRIC AVERAGE NOMINAL VELOCITY *
C     *   VLAVE     => VOLUMETRIC AVERAGE EFFECTIVE VELOCITY
C     *   BLOCK     => EFFECTIVE BLOCKAGE                  *
C     *                                                     *
C     *******************************************************
      OIMENSION VO(60,11),RX(11),UX(60),RA(11),UA(60,11),UASEG(60)
      DIMENSION RE(11), VR(60,11),VT(60,11),AUASEG(44),AUX(44),VTE(11)
      DIMENSION THETA(60),VE(60,11),IOENT(18),UERX(60),VRE(11)
      COMMON RX,UX,RA,UASEG,UERX,RE
  110 FORMAT(8F10.5)
  103 FORMAT(/' AVERAGE VELOCITY =',F5.3)
  113 FORMAT(/' VOLUMETRIC AVERAGE NOMINAL VELOCITY =',F5.3)
  114 FORMAT(/' VOLUMETRIC AVERAGE EFFECTIVE VELOCITY =',F5.3)
  115 FORMAT(/' EFFECTIVE BLOCKAGE =',F7.3,' PERCENT')
  111 FORMAT(18A4)
  210 FORMAT(16I5)
  200 FORMAT(8F10.5)
C
C *** REAO IN NOMINAL WAKE DATA.  SAVE RAOIAL ANO TANGENTIAL
```

```
C     VELDCITIES.  THEY ARE NOT NEEDEO FDR EFFECTIVE WAKE
C     CALCULATIONS BUT WILL BE ECHOED FDR WKPRDC.
C
      REAO(10,111) (IOENT(I),I=1,18)
      REAO(10,210) NX
      REAO(10,210) NP
      READ(10,200)(RX(M),M=1,NX)
      DO 20 M=1,NX
      REAO(10,200)((THETA(K),VD(K,M),VT(K,M),VR(K,M)),K=1,NP)
   20 CDNTINUE
C
C *** REAO IN PROPELLER-INDUCED VELOCITIES FROM EITHER FPUV
C     OR A SEPARATE OATA FILE
C
 900  FDRMAT(/' DO YOU WISH TD USE STEAOY OR UNSTEAOY VELDCITY PRDFILE?'
     &  /' ENTER 1 FOR STEADY, 2 FDR UNSTEADY')
   8  WRITE(6,900)
      READ(5,*) ISTED
      IF (ISTED.GT.2.DR.ISTEO.LT.1) GDTD 8
      REAO(8,210) NA
      REAO(8,200) (RA(I),I=1,NA)
      RH=RA(1)
      RPROP=1.
      IF(ISTEO.EQ.1) GDTD 22
      REAO(8,200) ((UA(KK,M),KK=1,NP),M=1,NA)
      GDTO 92
  22  REAO(8,200) (UA(1,M),M=1,NA)
      DO 21 KK=2,NP
      OO 21 M=1,NA
  21  UA(KK,M)=UA(1,M)
C
C *** EXTRAPOLATE NDMINAL WAKE DATA TO HUB
C
  92  WRITE(6,100)
  100 FDRMAT(/' OO YDU WISH TD EXTRAPDLATE VELDCITY DATA TD THE HUB?'/
     1  ' ENTER 2 FOR NO, 1 FOR LINEAR, O FDR CONSTANT')
      REAO(5,*)IXTRAP
      NX1=NX
      IF(IXTRAP.GT.1)GO TO 28
C
C *** LINEAR EXTRAPOLATION OF NDMINAL WAKE OATA TO HUB.
C
      NX=NX+1
      OD 25 K=1,NP
      OO 23 M=1,NX1
      MR=NX-M
      MRP1=MR+1
      VT(K,MRP1)=VT(K,MR)
      VR(K,MRP1)=VR(K,MR)
   23 VD(K,MRP1)=VO(K,MR)
   25 VO(K,1)=VD(K,2)+IXTRAP*(VO(K,2)-VO(K,3))/(RX(1)-RX(2))*(RH-RX(1))
      DO 26 M=1,NX1
      MR=NX-M
```

```
      MRP1=MR+1
   26 RX(MRP1)=RX(MR)
      RX(1)=RH
   28 VL=0.
      VLAVN=0.
      VLAVE=0.
C
C *** CYCLE THROUGH EACH PIE SEGMENT
C
      DO 50 K=1,NP
      DO 30 M=1,NX
      UASEG(M)=UA(K,M)
   30 UX(M)=VO(K,M)
C
C *** WKMOD PERFORMS THE HUANG MODIFICATION ON ONE PIE-SEGMENT
C
      CALL WKMOD(NX,NX,NA,1,RPROP,VOL,VOLAVN,VOLAVE)
C
C *** INTERPOLATE VT AND VR AT EFFECTIVE WAKE RADII
C
      DO 60 M=1,NX
      UASEG(M)=VT(K,M)
  60  UX(M)=VR(K,M)
      CALL UGLYDK(NX,1,1,RX,UASEG,0.,0.,AUASEG)
      CALL UGLYDK(NX,1,1,RX,UX,0.,0.,AUX)
      CALL EVALDK(NX,NX,RX,RE,VTE,AUASEG)
      CALL EVALDK(NX,NX,RX,RE,VRE,AUX)
      DO 70 M=1,NX
      VT(K,M)=VTE(M)
  70  VR(K,M)=VRE(M)
C
C *** ACCUMULATE VOLUMETRIC AVERAGE DATA
C
      VL=VL+VOL
      VLAVN=VLAVN+VOLAVN
      VLAVE=VLAVE+VOLAVE
      DO 40 M=1,NX
   40 VE(K,M)=UERX(M)
   50 CONTINUE
C
C *** COMPUTE VOLUMETRIC AVERAGES
C
      VL=VL/NP
      VLAVN=VLAVN/NP
      VLAVE=VLAVE/NP
      BLOCK=(1.-VLAVE/VLAVN)*100.
C
C *** WRITE EFFECTIVE WAKE DATA IN FILE 11 TO BE PROCESSED BY WKPROC
C
      WRITE(11,111) (IDENT(I),I=1,18)
      WRITE(11,210) NX
      WRITE(11,210) (NP,M=1,NX)
      WRITE(11,200)(RE(M),M=1,NX)
```

```
      DO 120 M=1,NX                                                     PIEO1600
      WRITE(11,200)((THETA(K),VE(K,M),VT(K,M),VR(K,M)),K=1,NP)          PIEO1610
  120 CONTINUE                                                          PIEO1620
C                                                                       PIEO1630
C *** WRITE AVERAGES TO TERMINAL                                        PIEO1640
C                                                                       PIEO1650
      WRITE(6,103)VL                                                    PIEO1660
      WRITE(6,113)VLAVN                                                 PIEO1670
      WRITE(6,114)VLAVE                                                 PIEO1680
      WRITE(6,115)BLOCK                                                 PIEO1690
      STOP                                                              PIEO1700
      END                                                               PIEO1710
```

```
      SUBROUTINE WKMOD(N,NX,NA,IOW,RPROP,VOL,VOLAVN,VOLAVE)
C
C     *******************************************************
C     *                                                     *
C     *   SUBROUTINE WKMOD CALCULATES THE THOMAS HUANG      *
C     *   CONTRACTION OF THE NOMINAL AND INDUCED WAKES      *
C     *   PASSED TO IT THROUGH COMMON.  IT ALSO COMPUTES    *
C     *   VOLUMETRIC AVERAGES.                              *
C     *                                                     *
C     *   NX          => NUMBER OF NOMINAL RADII            *
C     *   RH          => RADIUS OF HUB                      *
C     *   RT          => RADIUS OF OUTERMOST NOMINAL VELOCITY*
C     *   UX,UXR      => NOMINAL WAKE VELOCITIES            *
C     *   UE,UEXR     => EFFECTIVE WAKE VELOCITIES          *
C     *   UP          => APPARENT WAKE = EFF. WAKE + IND. WAKE
C     *   UA,UAR      => INDUCED VELOCITIES                 *
C     *   IOW = 1     => NO TUNNEL CORRECTIONS              *
C     *   ITER        => OPEN WATER ITERATION INDEX         *
C     *   ITERAT      => HUANG CORRECTION FACTOR INDEX      *
C     *   B,C,D,F     => HUANG CORRECTION FACTORS. SEE PAPER *
C     *   AUA,AUX,AUE=> SPLINE COEFF. FOR INDUCED, NOMINAL,*
C     *                  AND EFFECTIVE VELOCITY FIELDS, RESP *
C     *   RPROP       => RADIUS OF PROPELLER                *
C     *   RE(M)       => SPECIFIED EFFECTIVE WAKE RADII     *
C     *   VOL         => AVERAGE VELOCITY                   *
C     *   VLAVN       => VOLUMETRIC AVERAGE NOMINAL VELOCITY *
C     *   VLAVE       => VOLUMETRIC AVERAGE EFFECTIVE VELOCITY
C     *                                                     *
C     *******************************************************
      DIMENSION RX(11),R(11),RA(11),UX(60),UXR(60),UA(60),UP(60),UE(60)
      DIMENSION RE(11),UAR(60),RP(11)
      DIMENSION AUX(240),AUA(240)
      DIMENSION AUE(240),UERX(60)
      COMMON RX,UX,RA,UA,UERX,RE
C
C *** USE R(I) FOR NOMINAL RADII
C
      N=NX
      N1=N-1
      RH=RX(1)
      RT=RX(NX)
C
C *** INTERPOLATE TO FIND NOMINAL VELOCITIES AT NOMINAL RADII
C
      CALL UGLYDK(NX,1,1,RX,UX,0.,0.,AUX)
      DO 10 I=1,N
C  10 R(I)=(RT-RH)/(N-1)*(I-1)+RH
   10 R(I)=RX(I)
      CALL EVALDK(NX,N,RX,R,UXR,AUX)
C
C *** INTERPOLATE TO FIND INDUCED VELOCITIES AT NOMINAL RADII
C
      CALL UGLYDK(NA,1,1,RA,UA,0.,0.,AUA)
```

```
      CALL EVALOK(NA,N,RA,R,UAR,AUA)
 901  FORMAT(8F10.5)
C
C *** OUTSIOE PROPELLER OISK, NOMINAL WAKE = EFFECTIVE WAKE
C
      IUPDAT=0
      ITER=1
      ITERAT=1
      UE(N)=UXR(N)
   20 UP(N)=UE(N)+UAR(N)
C
C *** ESTIMATE EFF. WAKE MARCHING FROM OUTSIOE TO HUB
C
      DO 30 I=1,N1
      K=N-I
      UE(K)=SQRT((UE(K+1)+(UAR(K+1)+UAR(K))/2.)**2+UXR(K)**2-UXR(K+1)**2
     1)-(UAR(K+1)+UAR(K))/2.
      UP(K)=UE(K)+UAR(K)
   30 CONTINUE
C
C *** COMPUTE HUANG'S CORRECTION FACTORS
C
      RP(1)=RH
      VOL=0.
      OO 40 K=1,N1
      B=2*UP(K+1)+UP(K)
      C=-RP(K)*(UP(K+1)-UP(K))
      F=(R(K+1)**2-R(K)**2)*(2*UXR(K+1)+UXR(K))-R(K)*(UXR(K+1)-UXR(K))*(
     1R(K+1)-R(K))
      VOL=VOL+F
      D=-RP(K)**2*(UP(K+1)+2*UP(K))-F
C
C *** FINO RAOIUS FOR (K+1)ST APPARENT WAKE
C
      IF ((C*C-4*B*D).LT.0.) GOTO 80
   40 RP(K+1)=(-C+SQRT(C**2-4*B*D))/2./B
      IF(ABS(RP(N)-RT).LT.(.0001*RT))GO TO 60
      IUPOAT=0
C
C *** IOW=0 => TUNNEL CORRECTIONS
C        =1 => OPEN WATER (I.E., INFINITE FLUID)
C
      IF(IOW.EQ.1) GO TO 60
      IF (ITER.NE.1)GO TO 50
      ITER=2
      UENOLO=UE(N)
      RPNOLD=RP(N)
      UE(N)=UE(N)*.98
      GO TO 20
   50 CONTINUE
      ITER=ITER+1
      IF(ITER.GT.10)GO TO 80
      UEN=UE(N)
```

```
      UE(N)=UE(N)+(RT-RP(N))/(RP(N)-RPNOLD)*(UE(N)-UENOLD)              PIE02780
      RPNOLD=RP(N)                                                      PIE02790
      UENOLD=UEN                                                        PIE02800
      GO TO 20                                                          PIE02810
C                                                                       PIE02820
C *** ITERATE ON HUANG CORRECTION FACTORS, 3 TIMES FOR OPEN WATER       PIE02830
C                                                                       PIE02840
   60 IF(IUPDAT.EQ.1)GO TO 70                                           PIE02850
      IF(IDW.EQ.1.AND.ITERAT.EQ.5)GO TO 70                              PIE02860
      ITER=1                                                            PIE02870
      ITERAT=ITERAT+1                                                   PIE02880
      IF(ITERAT.GT.10)GO TO 80                                          PIE02890
      IUPDAT=1                                                          PIE02900
C                                                                       PIE02910
C *** FIND INDUCED VELOCITIES AT APPARENT WAKE RADII                    PIE02920
C                                                                       PIE02930
      CALL EVALDK(NA,N,RA,RP,UAR,AUA)                                   PIE02940
      GO TO 20                                                          PIE02950
   70 CONTINUE                                                          PIE02960
C                                                                       PIE02970
C *** CALCULATE EFFECTIVE WAKE RADII                                    PIE02980
C                                                                       PIE02990
      NXM1=NX-1                                                         PIE03000
      DO 100 M=1,NX                                                     PIE03010
  100 RE(M)=RH+(RPROP-RH)*FLOAT(M-1)/FLOAT(NXM1)                        PIE03020
C                                                                       PIE03030
C *** IF OUTERMOST APPARENT RADIUS IS INSIDE PROPELLER DISK, THEN       PIE03040
C     LINEARLY EXTRAPOLATE TO PROPELLER RADIUS                          PIE03050
C                                                                       PIE03060
      IF (RP(N).GE.RPROP) GOTO 110                                      PIE03070
      UE(N)=UE(N1) + (UE(N)-UE(N1))*(RPROP-R(N1))/(RP(N)-RP(N1))        PIE03080
      RP(N)=RPROP                                                       PIE03090
C                                                                       PIE03100
C *** INTERPOLATE EFFECTIVE WAKE VELOCITIES AT SPECIFIED RADII          PIE03110
C                                                                       PIE03120
  110 CALL UGLYDK(N,1,1,RP,UE,0.,0.,AUE)                                PIE03130
      CALL EVALDK(N,NX,RP,RE,UERX,AUE)                                  PIE03140
C                                                                       PIE03150
C *** COMPUTE VOL = AVERAGE VELOCITY                                    PIE03160
C                                                                       PIE03170
      VOL=VOL/3./(RT**2-RH**2)                                          PIE03180
C                                                                       PIE03190
C *** COMPUTE VOLUMETRIC AVERAGE NOMINAL VELOCITY                       PIE03200
C                                                                       PIE03210
      CALL INTEDK(NX,RX,RH,RPROP,YDX,XYDX,XXYDX,AUX)                    PIE03220
      VOLAVN=XYDX*2/(RPROP**2-RH**2)                                    PIE03230
C                                                                       PIE03240
C *** COMPUTE VOLUMETRIC AVERAGE EFFECTIVE VELOCITY                     PIE03250
C                                                                       PIE03260
      CALL INTEDK(N,RP,RH,RPROP,YDX,XYDX,XXYDX,AUE)                     PIE03270
      VOLAVE=XYDX*2./(RPROP**2-RH**2)                                   PIE03280
      RETURN                                                            PIE03290
C                                                                       PIE03300
```

```
C *** TUNNEL CORRECTION OR HUANG CORRECTION ITERATIONS FAIL TO CONVERGE PIE03310
C                                                                       PIE03320
   80 STOP                                                              PIE03330
      END                                                               PIE03340
```

REFERENCES

(1) Boswell, Robert J., "Design, Cavitation Performance, and Open-Water Performance of a Series of Research Skewed Propellers", Naval Research and Development Center Report No. 3339, 1971

(2) Valentine, D.T., and Chase, A., "Highly Skewed Propeller Design for a Naval Auxiliary Oiler (AO-177)", Naval Research and Development Center Report No. SPD-544-12, 1976

(3) Hendrican, A., and Remmers, K., "Powering and Cavitation Performance for a Naval Fleet Oiler, AO-177 Class (Model 5326 with Propeller 4677)", Naval Research and Development Center Report No. SPD-544-14, 1976

(4) Burrill, L.C. and Emerson, A., "Propeller Cavitation: Further Tests on 16-inch Propellers in the Kings College Cavitation Tunnel", Northeast Coast Institute of Engineers and Shipbuilders, Vol. 79, pp. 295-320, 1962-63

(5) Kerwin, J.E., and Lee, C.-S., "Prediction of Stedy and Unsteady Marine Propeller Performance by Numerical Lifting-Surface Theory", SNAME Transactions, Vol. 86, pp. 218-253, 1978

(6) Min, K.-S., "Numerical and Experimental Methods for

the Prediction of Field Point Velocities Around Propeller Blades", Department of Ocean Engineering of Massachusetts Institute of Technology Report No. 78-12, 1978

(7) Lee, C.-S., "Prediction of Steady and Unsteady Performance of Marine Propellers with or without Cavitation by Numerical Lifting-Surface Theory", Ph.D. Thesis, MIT, 1979

(8) Huang, T.T., et al., "Propeller/Stern/Boundary-Layer Interaction on Axisymmetric Bodies: Theory and Experiment", DTNSRDC Report No. 76-0113, 1976

(9) Huang, T.T., and Groves, N.C., "Effective Wake: Theory and Experiment", 13th Symposium on Naval Hydrodynamics, Tokyo, Japan, 1980

(10) Bjarne, E., "US Navy Oiler (AO-177 Class) - Model Tests in SSPA Cavitation Tunnel #2 - Design Propeller", Swedish Maritime Research Center, Goteborg, Sweden, DTNSRDC Report No. 2564-1, 1980

Thesis
J2725
c.1

Jamison

199476

Evaluation of a highly skewed propeller for a Naval auxiliary (AO-177).